(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 101939755 A
(43)申请公布日 2011.01.05

(21)申请号 200880122728.4

(22)申请日 2008.12.22

(30)优先权数据
2007-334490 2007.12.26 JP

(85)PCT申请进入国家阶段日
2010.06.28

(86)PCT申请的申请数据
PCT/JP2008/073285 2008.12.22

(87)PCT申请的公布数据
WO2009/081896 JA 2009.07.02

(71)申请人 CIS 电子工业有限公司
地址 巴西圣保罗

(72)发明人 伊豆山康夫

(74)专利代理机构 北京林达刘知识产权代理事务所（普通合伙）11277
代理人 刘新宇

(51)Int.Cl.
*G06K 17/00*(2006.01)
*G11B 5/09*(2006.01)
*G11B 5/10*(2006.01)
*G11B 20/10*(2006.01)

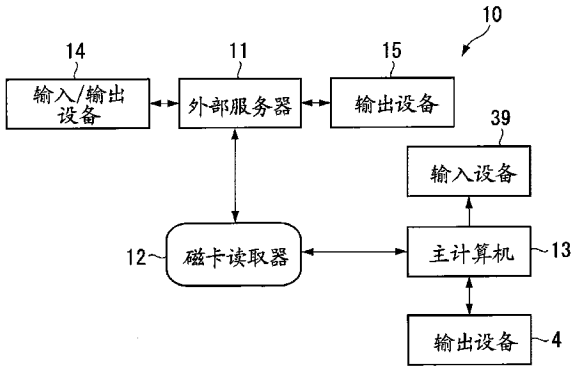
权利要求书 2 页 说明书 22 页 附图 23 页

(54)发明名称

磁头

(57)摘要

提供一种配置有能够存储从外部服务器下载的固件的微处理器的磁头。该磁头包括：芯，其具有将所述磁卡中所存储的数据转换成模拟信号的线圈；A/D 转换芯片，其连接至所述芯，并将所述模拟信号转换成数字信号；以及连接至所述 A/D 转换芯片的数字 IC。该处理器具有固件存储部件。当从外部服务器 (11) 向磁头下载控制该微处理器的计算和存储功能并控制外部硬件的固件时，固件存储部件存储该固件。




CN 101939755 A

1. 一种磁头，其从通过使用磁性材料存储有各种数据的磁卡读取所述数据，所述磁头包括：

芯，其具有将所述磁卡中所存储的数据转换成模拟信号的线圈；

A/D 转换芯片，其连接至所述芯，并将所述模拟信号转换成数字信号；以及

连接至所述 A/D 转换芯片的数字 IC，

其中，所述数字 IC 具有用于在从外部服务器向所述磁头下载固件时存储所述固件的固件存储部件，其中，所述固件控制所述数字 IC 的计算和存储功能并控制外部硬件。

2. 根据权利要求 1 所述的磁头，其特征在于，所述固件包括允许所述数字 IC 读取所述磁卡中的各种数据、从而支持所述磁卡的各种格式的数据读取控制，并且所述数字 IC 具有用于从所述磁卡读取各种数据、从而支持所述磁卡的各种格式的格式支持读取部件。

3. 根据权利要求 1 或 2 所述的磁头，其特征在于，所述固件包括允许所述数字 IC 基于预定加密算法对所述数字信号进行加密的数据加密控制，并且所述数字 IC 具有用于基于预定加密算法对所述数字信号进行加密的数据加密部件。

4. 根据权利要求 1 至 3 中任一项所述的磁头，其特征在于，所述数字 IC 具有用于在从所述外部服务器向所述磁头下载升级后的固件时、将版本升级前的固件重写为版本升级后的固件的固件更新部件。

5. 根据权利要求 1 至 4 中任一项所述的磁头，其特征在于，所述外部服务器使用存储在所述外部服务器中的密钥对所述固件进行加密，并将加密后的固件下载至所述磁头，并且所述数字 IC 使用存储在所述数字 IC 中的密钥对该加密后的固件进行解密，并存储解密后的固件。

6. 根据权利要求 1 至 5 中任一项所述的磁头，其特征在于，在所述数字 IC 和所述外部服务器进行相互认证、并判断为通过所述相互认证所获得的相互认证结果为正当之后，所述外部服务器向所述磁头下载所述固件，并且所述数字 IC 存储从所述外部服务器下载的固件，其中，在所述相互认证中，在所述数字 IC 和所述外部服务器之间进行认证。

7. 根据权利要求 1 至 6 中任一项所述的磁头，其特征在于，所述数字 IC 具有用于在从所述外部服务器向所述磁头下载对所述数字信号进行加密的各种加密算法时、存储所述加密算法的算法存储部件。

8. 根据权利要求 7 所述的磁头，其特征在于，所述数字 IC 具有用于在从所述外部服务器向所述磁头下载新加密算法时、将已经存储的加密算法重写为所述新加密算法的算法更新部件。

9. 根据权利要求 7 或 8 所述的磁头，其特征在于，所述外部服务器使用存储在所述外部服务器中的密钥对所述加密算法进行加密，并将加密后的加密算法下载至所述磁头，并且所述数字 IC 使用存储在所述数字 IC 中的密钥对该加密后的加密算法进行解密，并存储解密后的加密算法。

10. 根据权利要求 7 至 9 中任一项所述的磁头，其特征在于，在所述数字 IC 和所述外部服务器进行相互认证、并判断为通过所述相互认证所获得的相互认证结果为正当之后，所述外部服务器向所述磁头下载所述加密算法，并且所述数字 IC 存储从所述外部服务器下载的加密算法，其中，在所述相互认证中，在所述数字 IC 和所述外部服务器之间进行认证。

11. 根据权利要求 1 至 10 中任一项所述的磁头，其特征在于，所述磁头具有覆盖所述磁

头的外表面的壳体，并且所述芯、所述 A/D 转换芯片和所述数字 IC 容纳于所述壳体中。

12. 根据权利要求 1 至 11 中任一项所述的磁头，其特征在于，利用填充在所述壳体中的固态物质将所述 A/D 转换芯片和所述数字 IC 固定在所述壳体中。

# 磁头

## 技术领域

[0001] 本发明涉及从磁卡读取各种数据的磁头。

## 背景技术

[0002] 存在由磁头和连接至该磁头的主计算机形成的磁卡读取系统（参见专利文献 1）。该磁头由以下形成：头主体，其读取磁卡中所存储的数据；和控制单元，其将由头主体所读取的模拟信号转换成数字信号，并使用对称密钥加密方案或不对称密钥加密方案对该数字信号进行加密。头主体和控制单元容纳于头容器中。磁头的控制单元使用存储在其存储区域中的密钥对数字信号进行加密，并将加密后的数字信号发送至主计算机。主计算机的控制单元使用存储在其中的密钥对加密后的数字信号进行解密。

[0003] 在该系统中，当磁头的控制单元将加密后的数字信号发送至主计算机的控制单元时，该主计算机的控制单元指示磁头的控制单元改变密钥。该系统中用于改变密钥的过程如下所述。当主计算机的控制单元对从磁头接收到的数字信号进行解密时，该主计算机的控制单元生成新的密钥，并将所生成的密钥发送至磁头的控制单元。该磁头的控制单元将存储区域中所存储的现有密钥改变为新发送来的密钥。另外，当操作员通过键盘输入函数改变指令和新函数时，主计算机的控制单元将该函数改变指令和新函数发送至磁头的控制单元。磁头的控制单元将现有函数改变为新发送来的函数。

[0004] 专利文献 1：日本特开 2001-143213

## 发明内容

[0005] 发明要解决的问题

[0006] 在上述公报所公开的磁卡读取系统中的磁头的控制单元中，即使在磁头进入市场之后或在将磁头安装在磁卡读取器中之后、从外部服务器向该磁头下载对控制单元的计算和存储功能以及外部硬件进行控制的固件，由于该固件将不被存储在存储区域中，因此要下载的各种固件也不能够被存储在控制单元中。另外，即使从外部服务器下载升级后的固件，磁头的控制单元也不将版本升级后的固件存储在存储区域中，因而不能够支持固件升级。因此，当磁卡的格式改变时，磁头的控制单元可能不能够读取磁卡中的数据，在这种情况下，磁头自身需要随着格式的改变而变化。

[0007] 本发明的目的是提供一种包括能够存储从外部服务器下载的固件的数字 IC 的磁头。本发明的其它目的是提供包括能够存储从外部服务器下载的版本升级后的固件的数字 IC 的磁头。

[0008] 用于解决问题的方案

[0009] 本发明解决前述问题的前提是一种磁头，所述磁头从通过使用磁性材料存储有各种数据的磁卡读取所述数据。

[0010] 在本发明的该前提下，所述磁头包括：芯，其具有将所述磁卡中所存储的数据转换成模拟信号的线圈；A/D 转换芯片，其连接至所述芯，并将所述模拟信号转换成数字信号；

以及连接至所述 A/D 转换芯片的数字 IC，其中，所述数字 IC 具有用于在从外部服务器向所述磁头下载固件时存储所述固件的固件存储部件，其中，所述固件控制所述数字 IC 的计算和存储功能并控制外部硬件。

[0011] 作为本发明的例子，所述固件包括允许所述数字 IC 读取所述磁卡中的各种数据、从而支持所述磁卡的各种格式的数据读取控制，并且所述数字 IC 具有用于从所述磁卡读取各种数据、从而支持所述磁卡的各种格式的格式支持读取部件。

[0012] 作为本发明的另一例子，所述固件包括允许所述数字 IC 基于预定加密算法对所述数字信号进行加密的数据加密控制，并且所述数字 IC 具有用于基于预定加密算法对所述数字信号进行加密的数据加密部件。

[0013] 作为本发明的又一例子，所述数字 IC 具有用于在从所述外部服务器向所述磁头下载升级后的固件时、将版本升级前的固件重写为版本升级后的固件的固件更新部件。

[0014] 作为本发明的又一例子，所述外部服务器使用存储在所述外部服务器中的密钥对所述固件进行加密，并将加密后的固件下载至所述磁头，并且所述数字 IC 使用存储在所述数字 IC 中的密钥对该加密后的固件进行解密，并存储解密后的固件。

[0015] 作为本发明的又一例子，在所述数字 IC 和所述外部服务器进行相互认证、并判断为通过所述相互认证所获得的相互认证结果为正当之后，所述外部服务器向所述磁头下载所述固件，并且所述数字 IC 存储从所述外部服务器下载的固件，其中，在所述相互认证中，在所述数字 IC 和所述外部服务器之间进行认证。

[0016] 作为本发明的又一例子，所述数字 IC 具有用于在从所述外部服务器向所述磁头下载对所述数字信号进行加密的各种加密算法时、存储所述加密算法的算法存储部件。

[0017] 作为本发明的又一例子，所述数字 IC 具有用于在从所述外部服务器向所述磁头下载新加密算法时、将已经存储的加密算法重写为所述新加密算法的算法更新部件。

[0018] 作为本发明的又一例子，所述外部服务器使用存储在所述外部服务器中的密钥对所述加密算法进行加密，并将加密后的加密算法下载至所述磁头，并且所述数字 IC 使用存储在所述数字 IC 中的密钥对该加密后的加密算法进行解密，并存储解密后的加密算法。

[0019] 作为本发明的又一例子，在所述数字 IC 和所述外部服务器进行相互认证、并判断为通过所述相互认证所获得的相互认证结果为正当之后，所述外部服务器向所述磁头下载所述加密算法，并且所述数字 IC 存储从所述外部服务器下载的加密算法，其中，在所述相互认证中，在所述数字 IC 和所述外部服务器之间进行认证。

[0020] 作为本发明的又一例子，所述磁头具有覆盖所述磁头的外表面的壳体，并且所述芯、所述 A/D 转换芯片和所述数字 IC 容纳于所述壳体中。

[0021] 作为本发明的又一例子，利用填充在所述壳体中的固态物质将所述 A/D 转换芯片和所述数字 IC 固定在所述壳体中。

[0022] 发明的效果

[0023] 根据本发明的磁头，当从外部服务器向磁头下载控制数字 IC 的计算和存储功能、并控制外部硬件的固件时，数字 IC 存储该固件。由此，可以将在磁头进入市场之后或在将磁头安装在磁卡读取器中之后从外部存储器下载的固件随时存储在数字 IC 中。磁头即使在出厂或安装之后也可以支持各种固件，并且可以使用这种固件，根据磁头的工作环境进行数字 IC 的计算和存储功能以及外部硬件的最佳控制。

[0024] 在固件中包括允许数字 IC 读取磁卡中的各种数据从而支持磁卡的各种格式的数据读取控制、并且数字 IC 从磁卡读取各种数据从而支持磁卡的各种格式的磁头中，当从外部服务器向数字 IC 下载从磁卡读取数据从而支持磁卡的格式的数据读取控制时，数字 IC 存储该数据读取控制。由此，可以将在磁头进入市场之后或在将磁头安装在磁卡读取器中之后从外部服务器下载的数据读取控制随时存储在数字 IC 中。磁头即使在出厂或安装之后也可以支持磁卡的各种格式。因而，磁头可以适应于磁卡的各种规格，并且可以可靠地读取该卡中所存储的数据。磁头无需随着磁卡的格式的改变而变化，由此可以连续使用相同的磁头。

[0025] 在固件中包括允许数字 IC 基于预定加密算法对数字信号进行加密的数据加密控制、并且数字 IC 基于预定加密算法对数字信号进行加密的磁头中，数字 IC 对从磁卡读取的各种数据（数字信号）进行加密。因而，即使由第三方窃取了数据，除非对所窃取的数据进行解密，否则也不能够使用该数据，因此可以防止第三方对磁卡的未经授权的复制。注意，在网络银行中，存在窃取了磁卡中的数据的第三方进行在银行或信用卡公司的网站上创建伪网站的所谓的“电子欺骗”攻击、并在银行或信用卡公司处进行未经授权的交易的情况。然而，在该磁头中，由于第三方不能够窃取磁卡中的数据，因此第三方不能创建伪网站，从而使得能够防止第三方进行“电子欺骗”。

[0026] 即使在磁头进入市场之后或在将磁头安装在磁卡读取器中之后进行固件的升级，数字 IC 将版本升级前的固件重写为版本升级后的固件的磁头也可以立即支持版本升级后的固件。即使固件随着磁卡的格式的改变而变化，该磁头也可以存储改变后的固件，由此可以支持磁卡的各种改变后的格式。因此，该磁头可以适应于磁卡的各种规格，并且可以可靠地读取该卡中所存储的数据。

[0027] 在外部服务器使用存储在其中的密钥对固件进行加密、并且数字 IC 使用存储在其中的密钥对加密后的固件进行解密并存储解密后的固件的磁头中，将加密了的固件下载至磁头。因而，即使由第三方窃取了固件，除非对所窃取的固件进行解密，否则也不能够使用该固件，因此可以防止第三方通过固件窜改对磁卡的未经授权的使用。

[0028] 在数字 IC 和外部服务器进行在二者之间进行认证的相互认证并判断为通过相互认证所获得的相互认证结果为正当之后、外部服务器向磁头下载固件并且数字 IC 存储从外部服务器下载的固件的磁头中，数字 IC 和外部服务器可以通过进行相互认证判断相互正当性。因而，即使在伪服务器连接至磁头时或在伪磁头连接至外部服务器时，也可以检测到这种伪造。在该磁头中，第三方使用伪服务器不能够访问磁头，由此可以防止第三方通过固件窜改对磁卡的未经授权的使用。

[0029] 在从外部服务器下载加密算法时、数字 IC 存储该加密算法的磁头在进入市场之后或被安装至磁卡读取器中之后，可以将各种加密算法随时存储在数字 IC 中，并且可以使用各种加密算法对磁卡中的数据（数字信号）进行加密。在该磁头中，数字 IC 使用各种加密算法对磁卡中的数据进行加密。因而，即使由第三方窃取了数据，除非对所窃取的数据进行解密，否则不能够使用该数据。因此，可以可靠地防止第三方对磁卡中的数据的未经授权的获取，从而使得能够可靠地防止第三方对磁卡的未经授权的复制或第三方的“电子欺骗”。

[0030] 在从外部服务器下载新加密算法时、数字 IC 将已经存储的加密算法重写为该新

加密算法的磁头中，即使在磁头进入市场之后或在将磁头安装在磁卡读取器中之后分析数字 IC 中所存储的加密算法并由此需要改变算法，也可以立即处理加密算法的改变，由此可以基于改变后的加密算法对数据进行加密。在该磁头中，数字 IC 使用新加密算法对数据（数字信号）进行加密。因而，即使由第三方窃取了数据，除非对所窃取的数据进行解密，否则不能够使用该数据。因此，可以可靠地防止第三方对磁卡中的数据的未经授权的获取，从而使得能够可靠地防止第三方对磁卡的未经授权的复制或第三方的“电子欺骗”。

[0031] 在外部服务器使用存储在其中的密钥对加密算法进行加密、并且数字 IC 使用存储在其中的密钥对加密后的加密算法进行解密并存储解密后的加密算法的磁头中，将加密了的加密算法下载至该磁头中。因而，即使由第三方窃取了加密算法，除非对所窃取的加密算法进行解密，否则也不能够使用该加密算法来解密该数据。该磁头可以可靠地防止第三方对磁卡中的数据的未经授权的获取，由此可以可靠地防止第三方对磁卡的未经授权的复制或第三方的“电子欺骗”。

[0032] 在数字 IC 和外部服务器进行在二者之间进行认证的相互认证并判断为通过相互认证所获得的相互认证结果为正当之后、外部服务器向磁头下载加密算法并且数字 IC 存储从外部服务器下载的加密算法的磁头中，数字 IC 和外部服务器可以通过进行相互认证判断相互正当性。因而，即使伪磁头连接至外部服务器，该伪磁头也不能够访问外部服务器，因而没有从外部服务器向伪磁头下载加密算法。该磁头不允许第三方使用加密算法对磁卡中的数据进行解密，由此可以可靠地防止第三方对磁卡中的数据的未经授权的获取，从而使得能够可靠地防止第三方对磁卡的未经授权的复制或第三方的“电子欺骗”。

[0033] 在芯 (core)、A/D 转换芯片和数字 IC 容纳于壳体中的磁头中，除非磁头自身分解，否则转换成模拟信号或数字信号的数据不可能被窃取。因而，该磁头可以可靠地防止对存储在磁卡中的数据的窃取，由此可以可靠地防止第三方对磁卡的未经授权的复制或第三方的“电子欺骗”。

[0034] 在利用合成树脂将 A/D 转换芯片和数字 IC 固定于壳体中的磁头中，在磁头分解时需要去除合成树脂。当去除合成树脂时，A/D 转换芯片和数字 IC 被毁坏，从而使得能够防止数据窃取装置被安装在 A/D 转换芯片和数字 IC 上。该磁头可以可靠地防止第三方对磁卡中的数据的未经授权的获取，由此可以可靠地防止第三方对磁卡的未经授权的复制或第三方的“电子欺骗”。

## 附图说明

[0035] 图 1 是作为磁头的使用的例子示出的磁卡读取系统的硬件构成图。

[0036] 图 2 是作为例子示出的磁卡读取器的内部结构的示意图。

[0037] 图 3 是通过切除壳体的一部分示出的磁头的部分剖面立体图。

[0038] 图 4 是作为例子示出的微处理器（处理器）的构成图。

[0039] 图 5 是示出在外部服务器和磁头之间进行的处理的示例的框图。

[0040] 图 6 是示出外部认证的示例的梯形图。

[0041] 图 7 是示出内部认证的示例的梯形图。

[0042] 图 8 是示出在外部服务器和微处理器之间进行的下载处理的示例的梯形图。

[0043] 图 9 是用于说明加密和解密所使用的密钥的生成的示例的图。

[0044] 图 10 是用于说明加密和解密所使用的密钥的生成的示例的图。

[0045] 图 11 是用于说明加密和解密所使用的密钥的生成的示例的图。

[0046] 图 12 是用于说明加密和解密所使用的密钥的生成的示例的图。

[0047] 图 13 是用于说明加密和解密所使用的密钥的生成的示例的图。

[0048] 图 14 是用于说明加密和解密所使用的密钥的生成的示例的图。

[0049] 图 15 是示出在磁头和主计算机之间进行的处理的示例的框图。

[0050] 图 16 是示出外部认证的示例的梯形图。

[0051] 图 17 是示出内部认证的示例的梯形图。

[0052] 图 18 是示出系统中的主处理的示例的梯形图。

[0053] 图 19 是用于说明加密和解密所使用的密钥的生成的另一示例的图。

[0054] 图 20 是用于说明加密和解密所使用的密钥的生成的另一示例的图。

[0055] 图 21 是用于说明加密和解密所使用的密钥的生成的另一示例的图。

[0056] 图 22 是用于说明加密和解密所使用的密钥的生成的另一示例的图。

[0057] 图 23 是用于说明加密和解密所使用的密钥的生成的另一示例的图。

[0058] 图 24 是用于说明加密和解密所使用的密钥的生成的另一示例的图。

[0059] 附图标记说明

[0060] 10 :磁卡读取系统

[0061] 11 :外部服务器

[0062] 12 :磁卡读取器

[0063] 13 :主计算机

[0064] 19 :磁头

[0065] 23 :壳体

[0066] 24 :芯

[0067] 25 :A/D 转换芯片

[0068] 26 :微处理器（数字 IC）

[0069] 35 :中央处理单元

[0070] 36 :存储器

## 具体实施方式

[0071] 参考附图对根据本发明的磁头的详细说明如下所述。图 1 是作为磁头 19 的使用的例子示出的磁卡读取系统 10 的硬件构成图，并且图 2 是作为例子示出的磁卡读取器 12 的内部结构的示意图。图 3 是通过切除壳体 23 的一部分示出的磁头 19 的部分剖面立体图，并且图 4 是作为例子示出的微处理器 6（数字 IC）的构成图。在图 3 中，示出芯 24 的末端部 27 与磁卡 29 的表面相接触的状态，并且部分省略填充壳体 23 的合成树脂 28（固态物质）的图示。

[0072] 磁卡读取系统 10 由外部服务器 11、读取磁卡 29 中所存储的卡数据（各种数据）的磁卡读取器 12、和主计算机 13 形成。在系统 10 中，服务器 11 和卡读取器 12 经由接口（以有线或无线的方式）彼此连接，并且卡读取器 12 和计算机 13 经由接口（以有线或无线的方式）彼此连接。卡数据包括卡号、PIN、用户 ID、密码、卡持有者的个人信息（邮编、居住

地址或场所、姓名或称呼、出生日期、家庭结构、年收入、他 / 她所工作的公司、电话号码、传真号码、电子邮件地址和 URL 等）、卡持有者的法人信息（邮编、地址、姓名、成立日期、各种管理信息、客户信息、电话号码、传真号码、电子邮件地址和 URL 等）、以及商业交易内容等。

[0073] 外部服务器 11 是具有中央处理单元 (CPU 或 MPU) 和存储器（大容量硬盘）的计算机，并且具有 DNS 服务器功能。存储器存储卡读取器 12 的 URL。服务器 11 的中央处理单元由计算单元和控制单元形成。键盘和鼠标等的输入设备 14、以及显示器和打印机等的输出设备 15 经由接口连接至服务器 11。服务器 11 的中央处理单元基于操作系统的控制启动存储器中所存储的应用程序，并根据启动了的应用程序执行以下部件。

[0074] 外部服务器 11 的中央处理单元执行用于使用存储器中所存储的密钥对预定的固件进行加密的固件加密部件，并且执行用于使用存储器中所存储的密钥对预定的加密算法进行加密的算法加密部件。服务器 11 的中央处理单元执行用于经由因特网访问后面将说明的卡读取器 12 的控制器的访问部件，并且执行用于与磁头 19 相互进行认证的相互认证部件。

[0075] 外部服务器 11 的中央处理单元执行用于向磁头 19 下载未加密的固件或加密后的固件的第一固件下载部件，并且执行用于向磁头 19 下载新的未加密的固件（升级后的固件）或新的加密后的固件（升级后的固件）的第二固件下载部件。服务器 11 的中央处理单元执行用于向磁头 19 下载未加密的加密算法或加密后的加密算法的第一算法下载部件，并且执行用于向磁头 19 下载新的未加密的加密算法或新的加密后的加密算法的第二算法下载部件。

[0076] 固件是控制后面将说明的磁头 19 的微处理器 26 的计算和存储功能、并控制连接至处理器 26 的外部硬件的应用程序。固件包括允许磁头 19 的处理器 26 读取卡 29 中的各种数据、从而支持磁卡 29 的各种格式的数据读取控制。固件还包括允许磁头 19 的处理器 26 基于预定的加密算法对卡数据（数字信号）进行加密的数据加密控制。通过使用固件，该固件可以根据磁头 19 的工作环境进行处理器 26 的计算和存储功能的最佳控制，并进行连接至处理器 26 的外部硬件的最佳控制。

[0077] 磁卡读取器 12 是插入马达驱动型并且包括控制器（未示出）。如图 2 所示，卡读取器 12 具有形成于其前端处的卡插入口 16、形成于其后端处的卡排出口 17、和从卡插入口 16 连接至卡排出口 17 的卡导轨 18。在卡读取器 12 的中央处安装有磁头 19。在插入口 16、排出口 17 和磁头 19 附近安装用于检测在导轨 18 上移动的磁卡 29 的位置的光学传感器 20。

[0078] 当通过插入口 16 插入卡 29 时，卡 29 在导轨 18 上自动移动并且通过排出口 17 被排出。通过安装在卡读取器 12 中的带 21 使导轨 18 上的卡 29 移动。通过安装在卡读取器 12 中的马达 22 驱动带 21。磁头 19、传感器 20 和马达 22 连接至卡读取器 12 的控制器。

[0079] 卡读取器 12 的控制器是具有中央处理单元 (CPU 或 MPU) 和存储器（大容量闪速存储器）的计算机。存储器存储外部服务器 11 的 URL。控制器的中央处理单元由计算单元和控制单元形成。该控制器连接至 DNS 服务器（未示出）和主计算机 13。该控制器可以经由因特网访问外部服务器 11。该控制器通过接通 / 断开开关驱动或停止马达 22，并且向磁头 19 输出用于开始读取卡数据的指令或用于停止读取卡数据的指令。

[0080] 磁头 19 将磁卡 29 的磁层 32 上所存储的卡数据转换成电信号。如图 3 所示，磁头

19 由以下形成 :壳体 23,其覆盖磁头 19 的外表面 ;芯 24,其上安装有将磁卡中所存储的卡数据转换成模拟信号（电信号）的线圈（未示出）;A/D 转换芯片 25,其将模拟信号转换成数字信号（电信号）;以及微处理器 26(MPU)。在安装于卡读取器 12 中的磁头 19 中,形成磁头 19 的芯 24 的末端部 27 面向导轨 18。A/D 转换芯片 25 电连接至芯 24。处理器 26 电连接至 A/D 转换芯片 25,并且经由接口连接至主计算机 13。

[0081] 芯 24、A/D 转换芯片 25 和微处理器 26 容纳于壳体 23 中。然而,注意,芯 24 的末端部 27 从壳体 23 的下端暴露到外部。利用填充壳体 23 的合成树脂 28(固态物质）覆盖整个 A/D 转换芯片 25 和整个处理器 26,并利用合成树脂 28 将 A/D 转换芯片 25 和处理器 26 固定在壳体 23 内部。尽管对于合成树脂 28 优选使用热固性合成树脂,但除热固性合成树脂以外,还可以使用热塑性合成树脂。除合成树脂等的有机化合物以外,还可以使用陶瓷（固态物质）等对化学溶剂具有高耐性的无机化合物。在磁卡 29 中,将彩色印刷层 30、基层 31、磁层 32、遮蔽层 33 和印刷层 34 按该顺序从磁卡 29 的下侧开始排列。磁层 32 由铁磁性物质形成,并且基层 31 由聚对苯二甲酸乙二醇酯 (polyethylene terephthalate) 形成。注意,在磁头 19 上,代替微处理器 26,可以安装包括门阵列、现场可编程门阵列和专用硬件的数字 IC 中的任一个。

[0082] 如图 4 所示,微处理器 26 具有中央处理单元 35 和存储器 36(闪速存储器或 EEROM)。处理器 26 的中央处理单元 35 由计算单元 37 和控制单元 38 形成。中央处理单元 35 基于操作系统的控制启动存储器 36 中所存储的应用程序,并且根据启动了的应用程序执行以下部件。中央处理单元 35 执行用于与外部服务器 11 或主计算机 13 相互进行认证的相互认证部件。

[0083] 当从外部服务器 11 向磁头 19 下载未加密的固件时,微处理器 26 的中央处理单元 35 执行用于将固件存储在存储器 36 中的固件存储部件。可选地,当从服务器 11 向磁头 19 下载加密后的固件时,中央处理单元 35 执行用于使用存储器 36 中所存储的密钥对加密后的固件进行解密的固件解密部件,并且执行用于将解密后的固件存储在存储器 36 中的固件存储部件。当从服务器 11 向磁头 19 下载新的升级后的固件时,中央处理单元 35 执行用于将版本升级前的固件重写为版本升级后的固件的固件更新部件。

[0084] 当将固件存储在存储器 36 中时,微处理器 26 的中央处理单元 35 启动存储器中所存储的固件,并且根据启动了的固件执行以下部件。当从外部服务器 11 向磁头 19 下载加密前的各种加密算法时,中央处理单元 35 执行用于将加密算法存储在存储器 36 中的算法存储部件。可选地,当从外部服务器 11 向磁头 19 下载加密后的各种加密算法时,中央处理单元 35 执行用于使用存储器 36 中所存储的密钥对加密后的加密算法进行解密的算法解密部件,并且执行用于将解密后的加密算法存储在存储器 36 中的算法存储部件。

[0085] 当从外部服务器 11 向磁头 19 下载新的未加密的加密算法时,微处理器 26 的中央处理单元 35 执行用于将已经存储的加密算法重写为新加密算法的算法更新部件。可选地,当从外部服务器 11 向磁头 19 下载新的加密后的加密算法时,中央处理单元 35 执行用于使用存储器 36 中所存储的密钥对新的加密后的加密算法进行解密的算法解密部件,并且执行用于将已经存储的加密算法重写为新的解密后的加密算法的算法更新部件。中央处理单元 35 执行用于从卡 29 读取各种数据、从而支持磁卡 29 的各种格式的格式支持读取部件,并且执行用于基于预定的加密算法对卡数据（数据信号）进行加密的数据加密部件。中央

处理单元 35 执行用于将加密后的卡数据输出至主计算机 13 的加密数据输出部件。

[0086] 主计算机 13 具有中央处理单元 (CPU 或 MPU) 和存储器 ( 大容量硬盘 )。计算机 13 的中央处理单元由计算单元和控制单元形成。键盘和鼠标等的输入设备 39、以及显示器和打印机等的输出设备 40 经由接口连接至计算机 13。计算机 13 的中央处理单元基于操作系统的控制启动存储器中所存储的应用程序,并且根据启动了的应用程序执行以下部件。

[0087] 主计算机 13 的中央处理单元执行用于与磁头 19 的微处理器 26 相互进行认证的相互认证部件。当从磁头 19 输出加密后的卡数据时,计算机 13 的中央处理单元执行用于对数据进行解密的数据解密部件,并且执行用于将解密后的数据存储在存储器中的数据存储部件。计算机 13 的中央处理单元执行用于通过输出设备 40 输出解密后的数据的数据输出部件。注意,经由布线向外部服务器 11、磁卡读取器 12、主计算机 13、输入设备 14 和 39、以及输出设备 15 和 40 供电。

[0088] 图 5 是示出在外部服务器 11 和磁头 19 之间进行的处理的示例的框图。在外部服务器 11 和磁头 19 之间进行的相互认证的示例的说明如下所述。当启动系统 10 时,外部服务器 11、磁卡读取器 12 和主计算机 13 运行。服务器 11 使用卡读取器 12 的 URL 经由因特网访问卡读取器 12( 访问部件 )。可选地,卡读取器 12 使用服务器 11 的 URL 经由因特网访问服务器 11。

[0089] 当外部服务器 11 和磁卡读取器 12 的控制器经由因特网彼此连接时,服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 通过卡读取器 12 的控制器彼此连接。服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 进行存储器测试 (S-10) 和代码签名 (S-11)( 初始测试 )。在代码签名 (S-11) 中,判断是否已经重写了固件的对象代码。当初始测试结束并且其结果正确时,服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 进行用于判断它们的正当性的相互认证 ( 相互认证部件 )。在该相互认证中,服务器 11 进行用于认证磁头 19 的正当性的外部认证 (S-12),之后磁头 19 进行用于认证服务器 11 的正当性的内部认证 (S-13)。

[0090] 如果外部服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当,则使得能够从服务器 11 向磁头 19 下载固件或加密算法,由此在服务器 11 和处理器 26 之间进行下载处理 (S-14)。另一方面,如果服务器 11 和处理器 26 至少之一判断为认证结果为不正当,则在服务器 11 的显示器上显示认证不正当消息,由此不能够向磁头 19 下载固件或加密算法。

[0091] 可以在每次启动系统 10 时进行服务器 11 和处理器 26 之间的相互认证,或者可以在系统 10 连续运行时以日期和时间为单位、每周或每月进行服务器 11 和处理器 26 之间的相互认证,或者可以在每次向磁头 19 下载固件时进行服务器 11 和处理器 26 之间的相互认证,或者可以在每次向磁头 19 下载加密算法时进行服务器 11 和处理器 26 之间的相互认证。注意,在无需服务器 11 和处理器 26 进行相互认证的情况下,服务器 11 和处理器 26 经由因特网彼此连接、并且服务器 11 向磁头 19 下载固件或加密算法,这也是可以的。

[0092] 图 6 是示出外部认证的示例的梯形图,并且图 7 是示出内部认证的示例的梯形图。外部认证的认证过程如下所述。外部服务器的中央处理单元请求微处理器 26 的中央处理单元 35 生成并发送随机数 ( 认证符 )(S-20)。处理器 26 的中央处理单元 35 响应于来自服务器 11 的指令生成 64 位随机数,并将所生成的随机数发送至服务器 11(S-21)。已经获

得了 64 位随机数的服务器 11 的中央处理单元使用存储器中所存储的认证密钥，利用三重 DES(Data Encryption Standard，数据加密标准）对该随机数进行加密，之后将加密后的随机数发送至处理器 26(S-22)。处理器 26 的中央处理单元 35 使用存储器 36 中所存储的认证密钥，利用三重 DES 对加密后的随机数进行解密。处理器 26 的中央处理单元 35 将由其生成的随机数与解密后的随机数进行比较。如果这两个随机数相同，则中央处理单元 35 判断为认证结果为正当，并由此将认证结果正当信息发送至服务器 11(S-23)。另一方面，如果所生成的随机数和解密后的随机数不同，则中央处理单元 35 判断为认证结果为不正当，并由此将认证结果不正当信息发送至服务器 11(S-23)。服务器 11 从处理器 26 获得外部认证结果 (S-24)。

[0093] 在三重 DES 中，通过重复单次 DES（数据加密标准）三次来实现增加密钥长度和减少算法偏差，从而提高加密强度。三重 DES 包括所有三个密钥均不同的三密钥三重 DES、以及对于第一次加密和第三次加密使用相同的密钥的两密钥三重 DES。注意，三重 DES 可以是三密钥三重 DES 或两密钥三重 DES。还注意，代替三重 DES，该 DES 可以是单次 DES。

[0094] 内部认证的认证过程如下所述。外部服务器 11 的中央处理单元生成 64 位随机数（认证符）并将该 64 位随机数发送至微处理器 26(S-25)。已经获得了 64 位随机数的处理器 26 的中央处理单元 35 使用存储器 36 中所存储的认证密钥，利用三重 DES 对该随机数进行加密，之后将加密后的随机数发送至服务器 11(S-26)。服务器 11 的中央处理单元使用存储器中所存储的认证密钥，利用三重 DES 对加密后的随机数进行解密 (S-27)。服务器 11 的中央处理单元将由其生成的随机数与解密后的随机数进行比较。如果这两个随机数相同，则服务器 11 的中央处理单元判断为认证结果为正当。另一方面，如果所生成的随机数和解密后的随机数不同，则服务器 11 的中央处理单元判断为认证结果为不正当，并由此不允许将固件或加密算法下载至磁头 19。

[0095] 图 8 是示出在外部服务器 11 和微处理器 26 之间进行的下载处理的示例的梯形图。在外部服务器 11 的存储器中，存储有固件、加密算法、以及用于对固件和加密算法进行加密的加密密钥。在需要时，将新的升级后的固件或新的升级后的加密算法随时存储在外部服务器 11 的存储器中。在微处理器 26 的存储器 36 中，存储有用于对固件和加密算法进行解密的解密密钥。

[0096] 外部服务器 11 的中央处理单元从存储器提取固件、加密算法和加密密钥，并使用这些密钥利用三重 DES 对固件或加密算法进行加密（固件加密部件或算法加密部件）(S-28)。服务器 11 的中央处理单元经由因特网将加密后的固件或加密算法下载至磁头 19（第一固件下载部件或第一算法下载部件）(S-29)。注意，当固件或加密算法未被加密时，服务器 11 的中央处理单元将它们照原样下载至磁头 19，而没有进行加密（第一固件下载部件和第一算法下载部件）(S-29)。

[0097] 当外部服务器 11 的中央处理单元需要向磁头 19 下载新的升级后的固件或新的升级后的加密算法时，外部服务器 11 的中央处理单元从存储器提取新固件或新加密算法以及加密密钥，并使用这些密钥利用三重 DES 对该固件或该加密算法进行加密（固件加密部件或算法加密部件）(S-28)。服务器 11 的中央处理单元经由因特网向磁头 19 下载新的加密后的固件和新的加密后的加密算法（第二固件下载部件和第二算法下载部件）(S-29)。注意，当新固件或新加密算法未被加密时，服务器 11 的中央处理单元将它们照原样下载至

磁头 19，而没有进行加密（第二固件下载部件和第二算法下载部件）(S-29)。将从服务器 11 下载的固件或加密算法临时存储在磁卡读取器 19 的控制器的存储器中，之后从控制器将该所下载的固件或加密算法输出至磁头 19。

[0098] 当微处理器 26 的中央处理单元 35 从外部服务器 11 接收加密后的固件和加密算法时，中央处理单元 35 从存储器 36 提取解密密钥，并使用这些密钥利用三重 DES 对加密后的固件或加密算法进行解密（固件解密部件或算法解密部件）(S-30)。中央处理单元 35 将解密后的固件和加密算法存储在存储器 36 中（固件存储部件和算法存储部件）。当从外部服务器 11 接收到未加密的固件和未加密的加密算法时，中央处理单元 35 将该固件或加密算法存储在存储器 36 中（固件存储部件或算法存储部件）。

[0099] 当微处理器 26 的中央处理单元 35 从外部服务器 11 接收新的加密后的固件和新的加密后的加密算法时，中央处理单元 35 使用解密密钥利用三重 DES 对新的加密后的固件或新的加密后的加密算法进行解密（固件解密部件或算法解密部件）(S-30)。中央处理单元 35 将版本升级前的固件重写为版本升级后的解密后的固件（固件更新部件），并将版本升级后的固件存储在存储器 36 中。中央处理单元 35 进一步将已经存储的加密算法重写为新的解密后的加密算法（算法更新部件），并将该新加密算法存储在存储器 36 中。当从服务器 11 接收到新的未加密的固件或新的未加密的加密算法时，中央处理单元 35 将版本升级前的固件重写为版本升级后的固件（固件更新部件），并将该版本升级后的固件存储在存储器 36 中，并且将已经存储的加密算法重写为新加密算法（算法更新部件），并将该新加密算法存储在存储器 36 中。

[0100] 在升级固件时、或者在磁卡 29 的规格变化并由此卡 29 的格式改变时，进行固件的重写。在由于已经由第三方分析了算法因而产生重写的需求时，或者在每次启动系统 10 时，或者以日期和时间为单位、每周或每月，或者在同步丢失之后再次实现同步时，进行加密算法的重写。

[0101] 图 9 ～ 14 是用于说明加密和解密所使用的密钥的生成的示例的图。在每次将加密后的固件或加密后的加密算法下载至磁头 19 时，外部服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 使用预先存储在其存储器 36 中的相同且有限的回归计数值，彼此同步地依次生成相同的并且是对固件或加密算法进行加密和解密所需的新的第 2 密钥至第 n 密钥（密钥生成部件）。由服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 所进行的密钥生成处理的示例的说明如下所述。注意，回归计数值为 1 ～ 20。然而，注意，没有特别限制回归计数值，并且该计数值可以为 21 以上。

[0102] 当外部服务器 11 向磁头 19 下载第 1 个固件（新固件）或加密算法（新算法）时，如图 9 所示，服务器 11 的中央处理单元从存储器中所存储的计数表中选择回归计数值 1，并将计数值 1 添加至该固件或加密算法。在该计数表中，创建各个计数值 (1 ～ 20) 用的存储区域以及与各存储区域相关联的三个密钥存储区域 (K1、K2 和 K3)。然而，注意，在图 9 中的计数表中，没有生成与回归计数值 2 ～ 20 分别相关联的第 2 密钥至第 20 密钥。注意，在导入系统 10 时将与计数值 1 相关联的第 1 密钥（密钥 1）设置为初始值。

[0103] 外部服务器 11 的中央处理单元从计数表提取与计数值 1 相关联的第 1 密钥，使用这些第 1 密钥利用三重 DES（三密钥三重 DES）对固件或加密算法以及计数值 1 进行加密（固件加密部件或算法加密部件），并将加密后的固件或加密算法下载至磁头 19（第一固件

下载部件或第一算法下载部件）。在服务器 11 的中央处理单元将加密后的固件或加密算法下载至磁头 19 之后，服务器 11 的中央处理单元将回归计数值从 1 改变为 2，并将计数值 2 存储在存储器中。

[0104] 如图 10 所示，已经接收到加密后的固件（第 1 个固件）或加密后的加密算法（第 1 个加密算法）的微处理器 26 的中央处理单元 35 从存储器 36 中所存储的计数表中选择回归计数值 1。在该计数表中，创建各个计数值 (1 ～ 20) 用的存储区域以及与各存储区域相关联的三个密钥存储区域 (K1、K2 和 K3)。然而，注意，在图 10 中的计数表中，没有生成与计数值 2 ～ 20 分别相关联的第 2 密钥至第 20 密钥。注意，与计数值 1 相关联的第 1 密钥（密钥 1）与外部服务器 11 的存储器中所存储的第 1 密钥相同，并且在导入系统 10 时将该第 1 密钥设置为初始值。

[0105] 微处理器 26 的中央处理单元 35 从计数表提取与计数值 1 相关联的第 1 密钥，并且使用这些第 1 密钥利用三重 DES（三密钥三重 DES）对加密后的固件或加密算法进行解密，由此获得明文 (plaintext) 固件或明文算法（固件解密部件或算法解密部件）。在中央处理单元 35 对固件或加密算法进行解密之后，中央处理单元 35 将它们存储在存储器 36 中（固件存储部件和算法存储部件），将回归计数值从 1 改变为 2，并将计数值 2 存储在存储器 36 中。

[0106] 外部服务器 11 可以停止使用微处理器 26 当前所使用的固件或加密算法，从存储器中所存储的固件和加密算法中选择新固件或新算法，并允许微处理器 26 使用该固件或该算法。当服务器 11 向磁头 19 下载第 2 个固件（升级后的固件）或第 2 个加密算法（新加密算法）时，如图 11 所示，外部服务器 11 的中央处理单元从存储器中所存储的计数表中选择回归计数值 2，并将计数值 2 添加至第 2 个固件或加密算法。

[0107] 外部服务器 11 的中央处理单元生成通过使用单向散列函数对与计数值 1 相关联的第 1 密钥（初始值）和计数值 1 进行散列所获得的输出散列值，并使用该输出散列值作为与计数值 2 相关联的第 2 密钥（密钥 2）（密钥生成部件）。将作为第 2 密钥（密钥 2）的输出散列值写入计数表中与计数值 2 相关联的密钥存储区域 (K1、K2 和 K3)。注意，在图 11 中的计数表中，没有生成与回归计数值 3 ～ 20 分别相关联的第 3 密钥至第 20 密钥。

[0108] 外部服务器 11 的中央处理单元从计数表提取与计数值 2 相关联的第 2 密钥，使用这些第 2 密钥利用三重 DES（三密钥三重 DES）对固件或加密算法（包括计数值 2）进行加密（固件加密部件或算法加密部件），并将加密后的固件或加密算法下载至磁头 19（第二固件下载部件或第二算法下载部件）。在服务器 11 的中央处理单元将加密后的固件或加密算法下载至磁头 19 之后，服务器 11 的中央处理单元将回归计数值从 2 改变为 3，并将计数值 3 存储在存储器中。

[0109] 如图 12 所示，已经接收到加密后的固件（第 2 个固件）或加密算法（第 2 个加密算法）的微处理器 26 的中央处理单元 35 从存储器 36 中所存储的计数表中选择回归计数值 2。中央处理单元 35 生成通过使用单向散列函数对与计数值 1 相关联的第 1 密钥（初始值）和计数值 1 进行散列所获得的输出散列值，并使用该输出散列值作为与计数值 2 相关联的第 2 密钥（密钥 2）（密钥生成部件）。由中央处理单元 35 所使用的散列函数与由服务器 11 的中央处理单元所使用的散列函数相同，并且所生成的第 2 密钥（密钥 2）与由服务器 11 的中央处理单元所生成的第 2 密钥相同。将作为第 2 密钥（密钥 2）的输出散列值写

入计数表中与计数值 2 相关联的密钥存储区域 (K1、K2 和 K3)。注意，在图 12 中的计数表中，没有生成与回归计数值 3 ～ 20 分别相关联的第 3 密钥至第 20 密钥。

[0110] 微处理单元 26 的中央处理单元 35 从计数表提取与计数值 2 相关联的第 2 密钥，并且使用这些第 2 密钥利用三重 DES( 三密钥三重 DES) 对加密后的固件或加密算法进行解密，由此获得明文固件或明文算法 ( 固件解密部件或算法解密部件 )。在中央处理单元 35 对固件或加密算法进行解密之后，中央处理单元 35 将它们存储在存储器 36 中 ( 固件存储部件和算法存储部件 )，将回归计数值从 2 改变为 3，并将计数值 3 存储在存储器 36 中。

[0111] 当外部服务器 11 向磁头下载第 3 个固件 ( 升级后的固件 ) 或第 3 个加密算法 ( 新加密算法 ) 时，如图 13 所示，服务器 11 的中央处理单元从存储器中所存储的计数表中选择回归计数值 3，并将回归计数值 3 添加至第 3 个固件或第 3 个加密算法。

[0112] 外部服务器 11 的中央处理单元生成通过使用单向散列函数对与计数值 2 相关联的第 2 密钥 ( 密钥 2，散列值 ) 和计数值 2 进行散列所获得的输出散列值，并且使用该输出散列值作为与计数值 3 相关联的第 3 密钥 ( 密钥 3)( 密钥生成部件 )。将作为第 3 密钥 ( 密钥 3) 的输出散列值写入计数表中与计数值 3 相关联的密钥存储区域 (K1、K2 和 K3)。注意，在图 13 中的计数表中，没有生成与回归计数值 4 ～ 20 分别相对应的第 4 密钥至第 20 密钥。

[0113] 外部服务器 11 的中央处理单元从计数表提取与计数值 3 相关联的第 3 密钥，使用这些第 3 密钥利用三重 DES( 三密钥三重 DES) 对固件或加密算法 ( 包括计数值 3) 进行加密 ( 固件加密部件或算法加密部件 )，并将加密后的固件或加密算法下载至磁头 19( 第二固件下载部件或第二算法下载部件 )。在服务器 11 的中央处理单元将加密后的固件或加密算法下载至磁头 19 之后，服务器 11 的中央处理单元将回归计数值从 3 改变为 4，并将计数值 4 存储在存储器中。

[0114] 如图 14 所示，已经接收到加密后的固件 ( 第 3 个固件 ) 或加密算法 ( 第 3 个加密算法 ) 的微处理器 26 的中央处理单元 35 从存储器 36 中所存储的计数表中选择回归计数值 3。中央处理单元 35 生成通过使用单向散列函数对与计数值 2 相关联的第 2 密钥 ( 密钥 2) 和计数值 2 进行散列所获得的输出散列值，并使用该输出散列值作为与计数值 3 相关联的第 3 密钥 ( 密钥 3)( 密钥生成部件 )。由中央处理单元 35 生成的第 3 密钥 ( 密钥 3) 与由外部服务器 11 的中央处理单元生成的第 3 密钥相同。将作为第 3 密钥 ( 密钥 3) 的输出散列值写入计数表中与计数值 3 相关联的密钥存储区域 (K1、K2 和 K3)。注意，在图 14 中的计数表中，没有生成与回归计数值 4 ～ 20 分别相对应的第 4 密钥至第 20 密钥。

[0115] 微处理器 26 的中央处理单元 35 从计数表提取与计数值 3 相关联的第 3 密钥，并使用这些第 3 密钥利用三重 DES( 三密钥三重 DES) 对加密后的固件或加密算法进行解密，由此获得明文固件或明文算法 ( 固件解密部件或算法解密部件 )。在中央处理单元 35 对固件或加密算法进行解密之后，中央处理单元 35 将它们存储在存储器 36 中 ( 固件存储部件和算法存储部件 )，将回归计数值从 3 改变为 4，并将计数值 4 存储在存储器 36 中。

[0116] 这样，外部服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 依次使用回归计数值 1 ～ 20 并使用单向散列函数，彼此同步地生成第 2 密钥至第 n 密钥。当回归计数值超过 20 时，服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 再次使用计数值 1 以依次生成第 21 密钥至第 40 密钥。当服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 生成第 21 密钥时，服务器 11 的中央处理单元和中央处理单元 35 将相应的密

钥存储区域中所存储的第 1 密钥重写为第 21 密钥。当服务器 11 的中央处理单元和中央处理单元 35 生成第 22 密钥时，服务器 11 的中央处理单元和中央处理单元 35 将相应的密钥存储区域中所存储的第 2 密钥重写为第 22 密钥。

[0117] 在磁卡读取系统 10 中，服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 可以通过执行相互认证部件来判断相互正当性。因而，即使在伪服务器连接至磁头 19 时或者在伪磁头连接至外部服务器 11 时，也可以检测到这种伪造。在该系统 10 中，第三方使用伪服务器不能够访问磁头 19，由此可以防止第三方通过固件窜改对磁卡 29 的未经授权的使用。另外，在系统 10 中，由于第三方使用伪磁头不能够访问外部服务器 11，因此不会从服务器 11 向伪磁头下载加密算法。

[0118] 在系统 10 中，由于外部服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 单独生成第 2 密钥至第 n 密钥，因此无需从服务器 11 向处理器 26 发送密钥，从而使得能够防止在发送密钥的过程中对该密钥的未经授权的获取。在系统 10 中，服务器 11 的中央处理单元总是使用不同的密钥对固件或加密算法进行加密，并且处理器 26 的中央处理单元 35 总是使用不同的密钥对固件或加密算法进行解密。因而，即使由第三方获得了密钥，第三方也不能够对固件或加密算法进行解密。另外，由于对于第 2 密钥至第 n 密钥各自使用散列值，因此即使由第三方非法地获得了密钥，第三方也不能够破译该密钥，从而使得能够可靠地防止第三方对密钥的使用。

[0119] 在系统 10 中，外部服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 使用相同并且有限的回归计数值，彼此同步地依次生成第 2 密钥至第 n 密钥。因而，由服务器 11 生成的密钥和由处理器 26 生成的密钥可以彼此一致，从而使得能够防止由于所生成的密钥之间的不一致而导致不能够对加密数据进行解密。另外，由于通过对回归计数值进行散列所获得的输出散列值包括在作为第 2 密钥至第 n 密钥其中之一的输出散列值中，因此即使第三方非法地访问系统 10，第三方也不能够破译散列的回归计数值，因而不能够判断正在使用哪个计数值以进行服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 之间的同步。

[0120] 在系统 10 正在运行时外部服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 不同步时，由服务器 11 的中央处理单元生成的密钥不同于由处理器 26 的中央处理单元 35 生成的密钥，因而中央处理单元 35 不能够对从该中央处理单元下载的加密数据进行解密。在这种情况下，处理器 26 的中央处理单元 35 判断为不能够利用所生成的密钥进行解密，并由此向服务器 11 发送“不可解密”（“不可解密”信息发送部件）并请求与服务器 11 的再同步（再同步请求部件）。

[0121] 微处理器 26 的中央处理单元 35 请求卡读取器 12 的控制器访问外部服务器 11，并且使用存储器 36 中所存储的数据发送和接收密钥利用三重 DES 对“不可解密”信息和再同步请求进行加密。当服务器 11 和卡读取器 12 经由因特网彼此连接时，处理器 26 的中央处理单元 35 将加密后的“不可解密”信息和再同步请求发送至服务器 11。处理器 26 的中央处理单元 35 和已经接收到再同步请求的服务器 11 的中央处理单元进行用于判断它们的正当性的外部认证和内部认证（参见图 6 和 7）（相互认证部件）。如果服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当，则服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 使它们各自的回归计数值恢

复为 1（初始值）并再次开始同步。当服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 使它们各自的计数值恢复为 1 时，服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 再次使用第 1 密钥进行加密和解密。

[0122] 在系统 10 中，即使在由外部服务器 11 和微处理器 26 所生成的密钥之间发生不一致，服务器 11 和处理器 26 也可以使它们各自的回归计数值恢复为 1 并再次彼此同步。因而，由服务器 11 生成的密钥和由处理器 26 生成的密钥可以再次彼此一致，从而使得能够防止由于所生成的密钥之间的不一致而导致不能够对固件或加密算法进行解密。注意，在系统 10 连续运行、并由此以日期和时间为单位、每周或每月进行相互认证的情况下，当服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当时，服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 使它们各自的回归计数值恢复为 1 并再次开始同步。后续的过程步骤与基于图 9 ～ 14 所述的过程步骤相同。

[0123] 对于单向散列函数，使用 SHA-1(Secure Hash Algorithm 1，安全散列算法 1)、MD2、MD4、MD5(Message Digest 2、4、5，信息摘要 2、4、5)、RIPEMD-80、RIPEMD-128、RIPEMD-160 和 N 散列中的任一个。这些散列函数存储于外部服务器 11 的存储器和主计算机 13 的存储器中。

[0124] 对于加密算法，除 DES 以外，还可以使用 RSA、AES(Advanced Encryption Standard，高级加密标准)、IDEA(International Data Encryption Algorithm，国际数据加密算法)、FEAL-N/NX(Fast Encryption Algorithm，快速加密算法)、MULTI2(Multimedia Encryption2，多媒体加密 2)、MISTY、SXAL(Substitution Xor Algorithm，替换 Xor 算法)、MBAL(Multi BlockAlgorithm，多块算法)、RC2、RC5、ENCRiP、SAFFE(Secure AndFast Encryption Routine，安全快速加密例程)、Blowfish、Skipjack、Khufu、Khafre、CAST 和 GOST28147-89 中的任一个。这些算法存储于外部服务器 11 的存储器和主计算机 13 的存储器中。

[0125] 在系统 10 中，在无需外部服务器 11 和微处理器 26 进行图 9 ～ 14 所示的密钥生成的情况下，服务器 11 可以向处理器 26 下载固件或加密算法。该例子的说明如下所述。外部服务器 11 使用卡读取器 12 的 URL 经由因特网访问卡读取器 12（访问部件）。可选地，卡读取器 12 使用外部服务器 11 的 URL 经由因特网访问服务器 11。当服务器 11 和卡读取器 12 经由因特网彼此连接时，服务器 11 的中央处理单元和微处理器 26 的中央处理单元 35 通过控制器彼此连接。服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 进行用于判断它们的正当性的外部认证和内部认证（参见图 6 和 7)（相互认证部件）。如果服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当，则使得能够从服务器 11 向磁头 19 下载固件或加密算法，并由此在服务器 11 和处理器 26 之间进行下载处理。

[0126] 服务器 11 的中央处理单元使用存储器中所存储的信息发送和接收密钥利用三重 DES 对新固件或新加密算法进行加密（固件加密部件或算法加密部件），并将加密后的固件或加密算法下载至磁头 19（第一固件下载部件或第一算法下载部件）。从服务器 11 下载的固件或加密算法被临时存储在卡读取器 12 的控制器的存储器中，之后被输出至磁头 19。

[0127] 当微处理器 26 的中央处理单元 35 从服务器 11 接收加密后的固件或加密算法时，

中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对加密后的固件或算法进行解密，由此获得明文固件或明文算法（固件解密部件或算法解密部件），并将解密后的固件或算法存储在存储器中（固件存储部件或算法存储部件）。

[0128] 外部服务器 11 可以停止使用由微处理器 26 当前所使用的固件或加密算法，从存储器中所存储的固件和加密算法中选择新固件或新算法，并允许微处理器 26 使用该固件或算法。当服务器 11 允许处理器 26 使用新固件或新加密算法时，服务器 11 指示处理器 26 重写现有固件或现有加密算法（更新指令）。注意，假定已经进行了外部认证和内部认证（参见图 6 和 7），并且服务器 11 和处理器 26 已经判断为通过相互认证所获得的相互认证结果为正当。

[0129] 外部服务器 11 的中央处理单元使用存储器中所存储的信息发送和接收密钥利用三重 DES 对更新指令以及新固件或新加密算法进行加密（固件加密部件或算法加密部件），并将加密后的更新指令和加密后的固件或算法下载至磁头 19（第二固件下载部件或第二算法下载部件）。从服务器 11 下载的更新指令以及固件或加密算法被临时存储在卡读取器 12 的控制器的存储器中，之后被输出至磁头 19。

[0130] 当微处理器 26 的中央处理单元 35 从外部服务器 11 接收加密后的更新指令和加密后的固件或加密算法时，中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对加密后的更新指令以及加密后的固件或算法进行解密（固件解密部件或算法解密部件）。中央处理单元 35 将存储器 36 中所存储的现有固件重写为新的解密后的固件（固件更新部件），并将该新固件存储在存储器 36 中。中央处理单元 35 进一步将存储器 36 中所存储的现有算法重写为新的解密后的算法（算法更新部件），并将该新算法存储在存储器 36 中。中央处理单元 35 向服务器 11 通知更新完成（更新完成通知）。中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对更新完成通知进行加密，并将加密后的更新完成通知发送至服务器 11。

[0131] 外部服务器 11 可以停止使用当前所使用的散列函数，从存储器中所存储的散列函数中选择新散列函数，并使用该散列函数。可以在每次启动系统 10 时进行散列函数的改变，或者可以以日期和时间为单位、每周或每月进行散列函数的改变，或者可以在同步丢失之后再次实现同步时进行散列函数的改变。当服务器 11 使用新散列函数时，服务器 11 指示微处理器 26 重写现有散列函数（函数改变指令）。服务器 11 的中央处理单元访问卡读取器 12。当服务器 11 和卡读取器 12 经由因特网彼此连接时，服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 进行用于判断它们的正当性的外部认证和内部认证（参见图 6 和 7）（相互认证部件）。如果服务器 11 的中央处理单元和处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当，则服务器 11 的中央处理单元使用存储器中所存储的数据发送和接收密钥利用三重 DES 对函数改变指令和新散列函数进行加密，之后将加密后的函数改变指令和散列函数发送至处理器 26。

[0132] 当微处理器 26 的中央处理单元 35 接收函数改变指令和散列函数时，中央处理单元 35 使用存储器 36 中所存储的数据发送和接收密钥利用三重 DES 对加密后的函数改变指令和散列函数进行解密。处理器 26 的中央处理单元 35 将存储器 36 中所存储的现有散列函数改变为新的解密后的散列函数，之后向外部服务器 11 通知改变完成（改变完成通知部件）。中央处理单元 35 使用存储器 36 中所存储的数据发送和接收密钥利用三重 DES 对改

变完成通知进行加密，并将加密后的改变完成通知发送至服务器 11。在系统 10 中，由于以对函数改变指令和散列函数进行加密的方式进行散列函数的改变，因此第三方不会获得要使用的散列函数，从而使得能够防止散列函数被第三方破译。

[0133] 在磁卡读取系统 10 中，当从外部服务器 11 向磁头 19 下载固件或加密算法时，处理器 26 将固件或加密算法存储在存储器 36 中。由此，可以将在磁头 19 进入市场之后或在将磁头 19 安装在磁卡读取器 12 中之后从服务器 11 下载的固件或加密算法随时存储在处理器 26 中。

[0134] 在系统 10 中，即使在磁头 19 的出厂或安装之后，也可以支持各种固件。使用该固件，可以根据磁头 19 的工作环境进行处理器 26 的计算和存储功能以及外部硬件的最佳控制。在磁头 19 的出厂或安装之后，系统 10 也可以允许磁头 19 支持磁卡 29 的各种格式。由此，系统 10 可以允许磁头 19 适应于磁卡 29 的各种规格并可靠地读取卡 29 中所存储的数据。在系统 10 中，即使在磁头 19 的出厂或安装之后，也可以使用各种加密算法，并且可以使用这些算法对卡数据进行加密。在系统 10 中，处理器 26 将版本升级前的固件重写为版本升级后的固件。因而，即使在磁头 19 的出厂或安装之后对固件进行升级，也可以立即支持版本升级后的固件。

[0135] 图 15 是示出在磁头 19 和主计算机 13 之间进行的处理的示例的框图。当启动系统 10 时，主计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 进行存储器测试 (S-50) 和代码签名 (S-51)（初始测试）。当初始测试结束并且其结果为正确时，计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 进行用于判断它们的正当性的相互认证（相互认证部件）。在该相互认证时，计算机 13 进行用于认证磁头 19 的正当性的外部认证 (S-52)，之后磁头 19 进行用于认证计算机 13 的正当性的内部认证 (S-53)。

[0136] 如果计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当，则使得能够读取磁卡读取器 12 中的磁卡 29，并由此在计算机 13 和处理器 26 之间进行主处理 (S-54)。另一方面，如果计算机 13 和处理器 26 至少之一判断为认证结果为不正当，则不能够进行由卡读取器 12 对磁卡 29 的读取，并由此在计算机 13 的显示器上显示“不可读取”信息。不仅在每次启动系统 10 时进行相互认证，而且在系统 10 连续运行时以日期和时间为单位、每周或每月进行相互认证。另外，如后面将说明的，当在计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 之间发生同步不一致时，进行相互认证。

[0137] 图 16 是示出外部认证的示例的梯形图，并且图 17 是示出内部认证的示例的梯形图。外部认证的认证过程如下所述。主计算机 13 的中央处理单元请求微处理器 26 的中央处理单元 35 生成并发送随机数（认证符）(S-60)。处理器 26 的中央处理单元 35 响应于来自计算机 13 的指令生成 64 位随机数，并将所生成的随机数发送至计算机 13(S-61)。已经获得了 64 位随机数的计算机 13 的中央处理单元使用存储器中所存储的认证密钥利用三重 DES 对该随机数进行加密，之后将加密后的随机数发送至处理器 26(S-62)。

[0138] 微处理器 26 的中央处理单元 35 使用存储器 36 中所存储的认证密钥利用三重 DES 对加密后的随机数进行解密 (S-63)。处理器 26 的中央处理单元 35 将由其生成的随机数与解密后的随机数进行比较。如果这两个随机数相同，则中央处理单元 35 判断为认证结果为正当，并由此将认证结果正当信息发送至计算机 13。另一方面，如果所生成的随机数和解密

后的随机数不同，则中央处理单元 35 判断为认证结果不正当，并由此将认证结果不正当信息和表示磁卡 29 为不可读取的信息发送至计算机 13。计算机 13 从微处理器 26 获得外部认证结果 (S-64)。

[0139] 内部认证的认证过程如下所述。计算机 13 的中央处理单元生成 64 位随机数（认证符），并将该 64 位随机数发送至微处理器 26(S-65)。已经获得了 64 位随机数的处理器 26 的中央处理单元 35 使用存储器 36 中所存储的认证密钥利用三重 DES 对随机数进行加密，之后将加密后的随机数发送至计算机 13(S-66)。计算机 13 的中央处理单元使用存储器中所存储的认证密钥利用三重 DES 对加密后的随机数进行解密 (S-67)。计算机 13 的中央处理单元将由其生成的随机数与解密后的随机数进行比较。如果这两个随机数相同，则计算机 13 的中央处理单元判断为认证结果为正当。另一方面，如果所生成的随机数和解密后的随机数不同，则计算机 13 的中央处理单元判断为认证结果为不正当，并由此不允许读取卡读取器 12 中的磁卡 29。

[0140] 图 18 是示出系统 10 中的主处理的示例的梯形图。图 19 ～ 24 是用于说明加密和解密所使用的密钥的生成的其它示例的图。在由于相互认证的结果为正当因而使得能够读取磁卡 29 之后、卡持有者通过卡插入口 16 插入磁卡 29 时，驱动马达 22 并且卡 29 在导轨 18 上移动。当卡 29 通过插入口 16 时，相关的光学传感器 20 检测到卡 29，并且从光学传感器 20 输出卡插入信号，并将该卡插入信号输入至卡读取器 12 的控制器。当控制器接收卡插入信号时，该控制器向磁头 19 的微处理器 26 输出用于开始读取卡 29 中所存储的卡数据的指令。当磁卡 29 通过磁头 19 并且经由排出口 17 被排出时，相关的光学传感器 20 检测到磁卡 29，并且从光学传感器 20 输出卡通过信号，并将该卡通过信号输入至卡读取器 12 的控制器。当控制器接收卡通过信号时，该控制器向磁头 19 的处理器 26 输出用于停止读取卡数据的指令，并停止驱动马达 22。

[0141] 当磁卡 29 的磁性化的磁层 32 通过磁头 19 的芯 24 的末端部 27（芯 24 的间隙）时，在芯 24 中生成磁通，并且在与磁通连结的方向上生成诱导电动势，由此电流流过线圈。流过线圈的电流的值随着磁通的变化而改变。由线圈提取出磁卡 29 的磁层 32 上所存储的卡数据作为模拟信号，并将该模拟信号输入至连接至线圈的 A/D 转换芯片 25。A/D 转换芯片 25 将从线圈输入的模拟信号转换成数字信号。从 A/D 转换芯片 25 将该数字信号输入至微处理器 26，并存储在处理器 26 的存储器 36 中。

[0142] 当系统 10 正在运行时，主计算机 13 的中央处理单元以预定时间间隔询问处理器 26 在微处理器 26 的存储器 36 中是否存在要处理的卡数据（数据检查指令）。计算机 13 的中央处理单元使用存储器中所存储的信息发送和接收密钥利用三重 DES 对数据检查指令进行加密，并将加密后的数据检查指令发送至处理器 26(S-68)。注意，优选该预定时间间隔以秒为单位或以毫秒为单位。当处理器 26 的中央处理单元 35 接收该数据检查指令时，中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对加密后的数据检查指令进行解密。处理器 26 的中央处理单元 35 响应于来自计算机 13 的数据检查指令搜索存储器 36。如果将磁卡 29 的卡数据作为数字信号存储在存储器 36 中，则中央处理单元 35 向计算机 13 发送数据拥有作为应答（数据拥有信息）。如果存储器 36 中不存在卡数据，则中央处理单元 35 向计算机 13 发送数据未拥有作为应答（数据未拥有信息）。处理器 26 使用信息发送和接收密钥利用三重 DES 对数据拥有信息或数据未拥有信息进行

加密，并将加密后的数据拥有信息或数据未拥有信息发送至计算机 13(S-69)。

[0143] 当主计算机 13 的中央处理单元接收数据拥有信息或数据未拥有信息时，主计算机 13 的中央处理单元使用信息发送和接收密钥利用三重 DES 对该数据拥有信息或数据未拥有信息进行解密。当计算机 13 的中央处理单元接收数据未拥有信息时，计算机 13 的中央处理单元以预定间隔向微处理器 26 再次发送加密后的数据检查指令，以询问处理器 26 在存储器 36 中是否存在要处理的卡数据（数据检查指令）。当计算机 13 的中央处理单元接收数据拥有信息时，计算机 13 的中央处理单元请求处理器 26 发送处理器 26 的存储器 36 中所存储的卡数据（数据发送指令）。计算机 13 的中央处理单元使用信息发送和接收密钥利用三重 DES 对数据发送指令进行加密，并将加密后的数据发送指令发送至处理器 26(S-70)。当处理器 26 的中央处理单元 35 接收该数据发送指令时，中央处理单元 35 使用信息发送和接收密钥利用三重 DES 对加密后的数据发送指令进行解密。

[0144] 微处理器 26 的中央处理单元 35 从存储器 36 提取数字信号（卡数据）和加密密钥，并使用这些密钥对数字信号进行加密，由此获得加密数据（数据加密部件）(S-71)。中央处理单元 35 将该加密数据发送至主计算机 13（加密数据发送部件）。计算机 13 具有对加密数据进行放大的放大电路（未示出）。计算机 13 从存储器提取解密密钥，并使用这些密钥对由放大电路放大后的加密数据进行解密（数据解密部件）(S-72)。计算机 13 可以将解密后的数字信号（明文卡数据）作为字符信息显示在显示器上（数据输出部件），并且可以向打印机打印出解密后的数字信号（明文卡数据）作为打印信息（数据输出部件）。计算机 13 将加密后的数字信号或解密后的数字信号存储在存储器中（数据存储部件）。当计算机 13 对加密数据进行解密时，计算机 13 以预定间隔向处理器 26 再次发送加密后的数据检查指令，以询问处理器 26 在存储器 36 中是否存在要处理的卡数据（数据检查指令）。

[0145] 在每次将加密后的数字信号发送至计算机 13 时，计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 使用预先存储在存储器和存储器 36 中的相同且有限的回归计数值，彼此同步地依次生成相同的且是对数字信号进行加密和解密所需的新的第 2 密钥至第 n 密钥（密钥生成部件）。基于图 19 ～ 24 的、由计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 所进行的密钥生成过程的示例的说明如下所述。注意，回归计数值为 1 ～ 20。然而，注意，没有特别限制回归计数值，并且该计数值可以为 21 以上。

[0146] 在启动系统 10 之后，在从 A/D 转换芯片 25 向微处理器 26 输入第 1 个数字信号（卡数据）并将该数字信号存储在存储器 36 中之后、接收到数据发送指令时，如图 19 所示，处理器 26 的中央处理单元 35 从存储器 36 中所存储的计数表中选择回归计数值 1，并将计数值 1 添加至该数字信号。在该计数表中，创建各个计数值 (1 ～ 20) 用的存储区域以及与各存储区域相关联的三个密钥存储区域 (K1、K2 和 K3)。然而，注意，在图 19 中的计数表中，没有生成与回归计数值 2 ～ 20 分别相对应的第 2 密钥至第 20 密钥。注意，在导入系统 10 时将与计数值 1 相关联的第 1 密钥（密钥 1）设置为初始值。

[0147] 微处理器 26 的中央处理单元 35 从计数表提取与计数值 1 相关联的第 1 密钥，使用这些第 1 密钥利用三重 DES（三密钥三重 DES）对数字信号和计数值 1 进行加密（数据加密部件），并将加密数据发送至计算机 13（数据发送部件）。在处理器 26 的中央处理单元 35 将加密数据发送至计算机 13 之后，中央处理单元 35 将回归计数值从 1 改变为 2，并将计数值 2 存储在存储器 36 中，并从存储器 36 删除第 1 个数字信号（卡数据）。

[0148] 如图 20 所示，已经接收到第 1 个加密数据的主计算机 13 的中央处理单元从存储器中所存储的计数表中选择回归计数值 1。在该计数表中，创建各个计数值 (1 ～ 20) 用的存储区域以及与各存储区域相关联的三个密钥存储区域 (K1、K2 和 K3)。然而，注意，在图 20 中的计数表中，没有生成与回归计数值 2 ～ 20 分别相对应的第 2 密钥至第 20 密钥。注意，与计数值 1 相关联的第 1 密钥 ( 密钥 1) 与微处理器 26 的存储器 36 中所存储的第 1 密钥相同，并且在导入系统 10 时将该第 1 密钥设置为初始值。计算机 13 的中央处理单元从计数表提取与计数值 1 相关联的第 1 密钥，并使用这些第 1 密钥利用三重 DES( 三密钥三重 DES) 对加密数据进行解密，由此获得数字信号 ( 明文卡数据 )。在计算机 13 的中央处理单元对加密数据进行解密之后，计算机 13 的中央处理单元将回归计数值从 1 改变为 2，并将计数值 2 存储在存储器中。

[0149] 在从 A/D 转换芯片 25 将第 2 个数字信号 ( 卡数据 ) 输入至微处理器 26 并将该数字信号存储在存储器 36 中之后、接收到数据发送指令时，如图 21 所示，处理器 26 的中央处理单元 35 从存储器 36 中所存储的计数表中选择回归计数值 2，并将回归计数值 2 添加至该数字信号。处理器 26 的中央处理单元 35 生成通过使用单向散列函数对与计数值 1 相关联的第 1 密钥 ( 初始值 ) 和计数值 1 进行散列所获得的输出散列值，并使用该输出散列值作为与计数值 2 相关联的第 2 密钥 ( 密钥 2)( 密钥生成部件 )。将作为第 2 密钥 ( 密钥 2) 的输出散列值写入计数表中与计数值 2 相关联的密钥存储区域 (K1、K2 和 K3)。注意，在图 21 中的计数表中，没有生成与回归计数值 3 ～ 20 分别相对应的第 3 密钥至第 20 密钥。

[0150] 微处理器 26 的中央处理单元 35 从计数表提取与计数值 2 相关联的第 2 密钥，使用这些第 2 密钥利用三重 DES( 三密钥三重 DES) 对数字信号 ( 包括计数值 2) 进行加密，由此获得加密数据 ( 数据加密部件 )，并将该加密数据发送至计算机 13。在处理器 26 的中央处理单元 35 将该加密数据发送至计算机 13 之后，中央处理单元 35 将回归计数值从 2 改变为 3，并将计数值 3 存储在存储器 36 中，并且从存储器 36 删除第 2 个数字信号 ( 卡数据 )。

[0151] 如图 22 所示，已经接收到第 2 个加密数据的计算机 13 从存储器中所存储的计数表中选择回归计数值 2。计算机 13 的中央处理单元生成通过使用单向散列函数对与计数值 1 相关联的第 1 密钥 ( 初始值 ) 和计数值 1 进行散列所获得的输出散列值，并使用该输出散列值作为与计数值 2 相关联的第 2 密钥 ( 密钥 2)( 密钥生成部件 )。由计算机 13 的中央处理单元所使用的散列函数与由微处理器 26 的中央处理单元 35 所使用的散列函数相同，并且所生成的第 2 密钥 ( 密钥 2) 与由处理器 26 的中央处理单元 35 生成的第 2 密钥相同。将作为第 2 密钥 ( 密钥 2) 的输出散列值写入计数表中与计数值 2 相关联的密钥存储区域 (K1、K2 和 K3)。注意，在图 22 中的计数表中，没有生成与回归计数值 3 ～ 20 分别相对应的第 3 密钥至第 20 密钥。计算机 13 的中央处理单元从计数表提取与计数值 2 相关联的第 2 密钥，并使用这些第 2 密钥利用三重 DES( 三密钥三重 DES) 对加密数据进行解密，由此获得数字信号 ( 明文卡数据 )。在计算机 13 的中央处理单元对加密数据进行解密之后，计算机 13 的中央处理单元将回归计数值从 2 改变为 3，并将计数值 3 存储在存储器中。

[0152] 从 A/D 转换芯片 25 向微处理器 26 输入第 3 个数字信号 ( 卡数据 ) 并将该数字信号存储在存储器 36 中之后、接收到数据发送指令时，如图 23 所示，处理器 26 的中央处理单元 35 从存储器 36 中所存储的计数表中选择回归计数值 3，并将回归计数值 3 添加至该数字信号。处理器 26 的中央处理单元 35 生成通过使用单向散列函数对与计数值 2 相关联的第

2 密钥（密钥 2，散列值）和计数值 2 进行散列所获得的输出散列值，并使用该输出散列值作为与计数值 3 相关联的第 3 密钥（密钥 3）（密钥生成部件）。将作为第 3 密钥（密钥 3）的输出散列值写入计数表中与计数值 3 相关联的密钥存储区域（K1、K2 和 K3）。注意，在图 23 中的计数表中，没有生成与回归计数值 4 ～ 20 分别相对应的第 4 密钥至第 20 密钥。

[0153] 微处理器 26 的中央处理单元 35 从计数表提取与计数值 3 相关联的第 3 密钥，使用这些第 3 密钥利用三重 DES（三密钥三重 DES）对数字信号进行加密（包括计数值 3），由此获得加密数据（加密部件），并将该加密数据发送至计算机 13。在处理器 26 的中央处理单元 35 将该加密数据发送至计算机 13 之后，中央处理单元 35 将回归计数值从 3 改变为 4，并将计数值 4 存储在存储器 36 中，并且从存储器 36 删除第 3 个数字信号（卡数据）。

[0154] 如图 24 所示，已经接收到第 3 个加密数据的主计算机 13 的中央处理单元从存储器中所存储的计数表中选择回归计数值 3。计算机 13 的中央处理单元生成通过使用单向散列函数对与计数值 2 相关联的第 2 密钥（密钥 2）和计数值 2 进行散列所获得的输出散列值，并使用该输出散列值作为与计数值 3 相关联的第 3 密钥（密钥 3）（密钥生成部件）。由计算机 13 的中央处理单元生成的第 3 密钥（密钥 3）与由微处理器 26 的中央处理单元 35 生成的第 3 密钥相同。将作为第 3 密钥（密钥 3）的输出散列值写入计数表中与计数值 3 相关联的密钥存储区域（K1、K2 和 K3）。注意，在图 24 中的计数表中，没有生成与回归计数值 4 ～ 20 分别相对应的第 4 密钥至第 20 密钥。计算机 13 的中央处理单元从该计数表提取与计数值 3 相关联的第 3 密钥，并使用这些第 3 密钥利用三重 DES（三密钥三重 DES）对加密数据进行解密，由此获得数字信号（明文卡数据）。在计算机 13 的中央处理单元对加密数据进行解密之后，计算机 13 的中央处理单元将回归计数值从 3 改变为 4，并将计数值 4 存储在存储器中。

[0155] 这样，主计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 依次使用回归计数值 1 ～ 20 并使用单向散列函数，彼此同步地生成第 2 密钥至第 n 密钥。当回归计数值超过 20 时，计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 再次使用计数值 1 以依次生成第 21 密钥至第 40 密钥。当计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 生成第 21 密钥时，计算机 13 的中央处理单元和中央处理单元 35 将相应的密钥存储区域中所存储的第 1 密钥重写为第 21 密钥。当计算机 13 的中央处理单元和中央处理单元 35 生成第 22 密钥时，计算机 13 的中央处理单元和中央处理单元 35 将相应的密钥存储区域中所存储的第 2 密钥重写为第 22 密钥。

[0156] 在磁卡读取系统 10 中，主计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 可以通过执行相互认证部件来判断相互正当性。因而，即使在伪计算机连接至磁头 19 时或在伪磁头连接至计算机 13 时，也可以检测到这种伪造。在系统 10 中，第三方使用伪计算机或伪磁头不能够访问系统 10，由此可以防止磁卡 29 的卡数据、散列函数和密钥被窃取。

[0157] 在系统 10 中，在计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 判断为通过认证部件所获得的认证结果为正当时，处理器 26 的中央处理单元 35 执行数据加密部件和数据发送部件，并且计算机 13 的中央处理单元执行解密部件。因而，与在不进行认证的情况下进行这种部件的情况相比较，可以可靠地防止磁卡 29 中所存储的卡数据被窃取，从而使得能够可靠地防止由第三方对磁卡 29 的未经授权的复制或第三方的“电子欺骗”。

[0158] 在系统 10 中，由于主计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 单独生成第 2 密钥至第 n 密钥，因此无需从计算机 13 向处理器 26 发送密钥，从而使得能够防止在发送密钥的处理中对该密钥的未经授权的获取。在系统 10 中，处理器 26 的中央处理单元 35 总是使用不同的密钥进行加密，并且计算机 13 的中央处理单元总是使用不同的密钥进行解密。因而，即使由第三方获得了密钥，该第三方也不能够对磁卡 29 中所存储的卡数据进行解密。另外，由于对于第 2 密钥至第 n 密钥各自使用散列值，因此即使由第三方非法地获得密钥，第三方也不能破译该密钥，从而使得能够可靠地防止第三方对密钥的使用。

[0159] 在系统 10 中，主计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 使用相同并且有限的回归计数值彼此同步地依次生成第 2 密钥至第 n 密钥。因而，由计算机生成的密钥和由处理器 26 生成的密钥可以彼此一致，从而使得能够防止由于所生成的密钥之间的不一致而导致不能够对加密数据进行加密。另外，由于通过对回归计数值进行散列所获得的输出散列值包括在作为第 2 密钥至第 n 密钥其中之一的输出散列值中，因此即使第三方非法地访问系统 10，该第三方也不能够破译散列的回归计数值，因而不能够判断正在使用哪个计数值以进行计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 之间的同步。

[0160] 在系统 10 正在运行时主计算机 13 的中央处理单元和微处理器 26 的中央处理单元 35 不同步时，由计算机 13 的中央处理单元生成的密钥不同于由处理器 26 的中央处理单元 35 生成的密钥，因而计算机 13 的中央处理单元不能够对从中央处理单元 35 发送来的加密数据进行解密。在这种情况下，计算机 13 的中央处理单元判断为不能够利用所生成的密钥进行解密，并由此通知“不可解密”（“不可解密”信息）并请求再同步（再同步请求）。计算机 13 的中央处理单元使用存储器中所存储的信息发送和接收密钥利用三重 DES 对“不可解密”信息和再同步请求进行加密，并将加密后的“不可解密”信息和再同步请求发送至处理器 26。计算机 13 的中央处理单元和已经接收到再同步请求的处理器 26 的中央处理单元 35 进行用于判断它们的正当性的外部认证和内部认证（参见图 16 和 17）（相互认证部件）。如果计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当，则计算机 13 的中央处理单元和中央处理单元 35 使它们各自的回归计数值恢复为 1（初始值）并再次开始同步。当计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 使它们各自的回归计数值恢复为 1 时，计算机 13 的中央处理单元和中央处理单元 35 再次使用第 1 密钥进行加密和解密。

[0161] 在系统 10 中，即使在所生成的密钥之间发生不一致，主计算机 13 和微处理器 26 也可以使它们各自的回归计数值恢复为 1，并且再次彼此同步。因而，由计算机 13 生成的密钥和由处理器 26 生成的密钥可以彼此一致，从而使得能够防止由于所生成的密钥之间的不一致而导致不能够对卡数据进行解密。注意，在系统 10 连续运行、并由此每天、每周或每月进行相互认证的情况下，当计算机 13 的中央处理单元和处理器 26 的中央处理单元 35 判断为通过相互认证所获得的相互认证结果为正当时，计算机 13 的中央处理单元和中央处理单元 35 使它们各自的回归计数值恢复为 1，并再次开始同步。后续的过程步骤与基于图 19 ～ 24 所述的过程步骤相同。

[0162] 主计算机 13 可以停止使用当前所使用的散列函数，从存储器中所存储的散列函数中选择新散列函数，并使用该散列函数。可以在每次启动系统 10 时进行散列函数的改

变，或者可以每天、每周或每月进行散列函数的改变，或者可以在同步丢失之后再次实现同步时进行散列函数的改变。当计算机 13 使用新散列函数时，计算机 13 指示微处理器 26 重写现有散列函数（函数改变指令）。计算机 13 的中央处理单元使用存储器中所存储的信息发送和接收密钥利用三重 DES 对函数改变指令和新散列函数进行加密，并将加密后的函数改变指令和散列函数发送至处理器 26。

[0163] 当微处理器 26 的中央处理单元 35 接收函数改变指令和散列函数时，中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对加密后的函数改变指令和散列函数进行解密。处理器 26 的中央处理单元 35 将存储器 36 中所存储的现有散列函数改变为新的解密后的散列函数，之后向计算机 13 通知改变完成（改变完成通知）。中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对改变完成通知进行加密，并将加密后的改变完成通知发送至计算机 13。在系统 10 中，由于以对函数改变指令和散列函数进行加密的方式进行散列函数的改变，因此第三方没有获得要使用的散列函数，从而使得能够防止散列函数被第三方破译。

[0164] 主计算机 13 可以停止使用当前所使用的加密算法，从存储器中所存储的加密算法中选择新算法，并使用该算法。可以在在每次启动系统 10 时进行加密算法的改变，或者可以每天、每周或每月进行加密算法的改变，或者可以在同步丢失之后再次实现同步时进行加密算法的改变。当计算机使用新加密算法时，计算机 13 指示微处理器 26 重写现有算法（函数改变指令）。计算机 13 的中央处理单元使用存储器中所存储的信息发送和接收密钥利用三重 DES 对函数改变指令和新加密算法进行加密，并将加密后的函数改变指令和算法送至处理器 26。

[0165] 在微处理器 26 的中央处理单元 35 接收函数改变指令和加密算法时，中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对加密后的函数改变指令和算法进行解密。处理器 26 的中央处理单元 35 将存储器 36 中所存储的现有算法改变为新的解密后的算法，之后向计算机 13 通知改变完成（改变完成通知）。中央处理单元 35 使用存储器 36 中所存储的信息发送和接收密钥利用三重 DES 对改变完成通知进行加密，并将加密后的改变完成通知发送至计算机 13。在系统 10 中，由于以对函数改变指令和加密算法进行加密的方式进行算法的改变，因此第三方没有获得要使用的散列函数。

[0166] 对于系统 10 中的磁卡读取器，除插入马达驱动型的卡读取器以外，还可以使用其上安装有磁头 19 的易操作的手动卡读取器。另外，可以将磁卡读取器连接至 POS 系统。当将磁卡读取器连接至 POS 系统时，无需在该卡读取器中安装控制器，并且利用 POS 系统中所包括的计算机建立到外部服务器 11 的连接。从外部服务器向 POS 系统的计算机的存储器下载固件或加密算法，并将该固件或加密算法临时存储在存储器中，之后输出至卡读取器的磁头的微处理器 26。

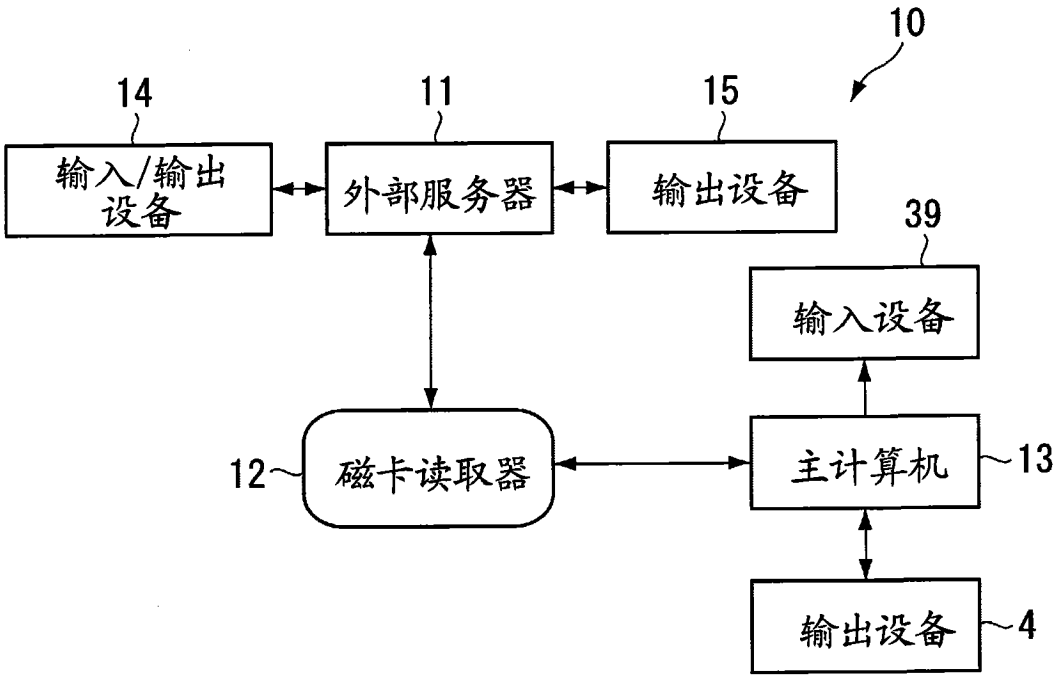



图 1

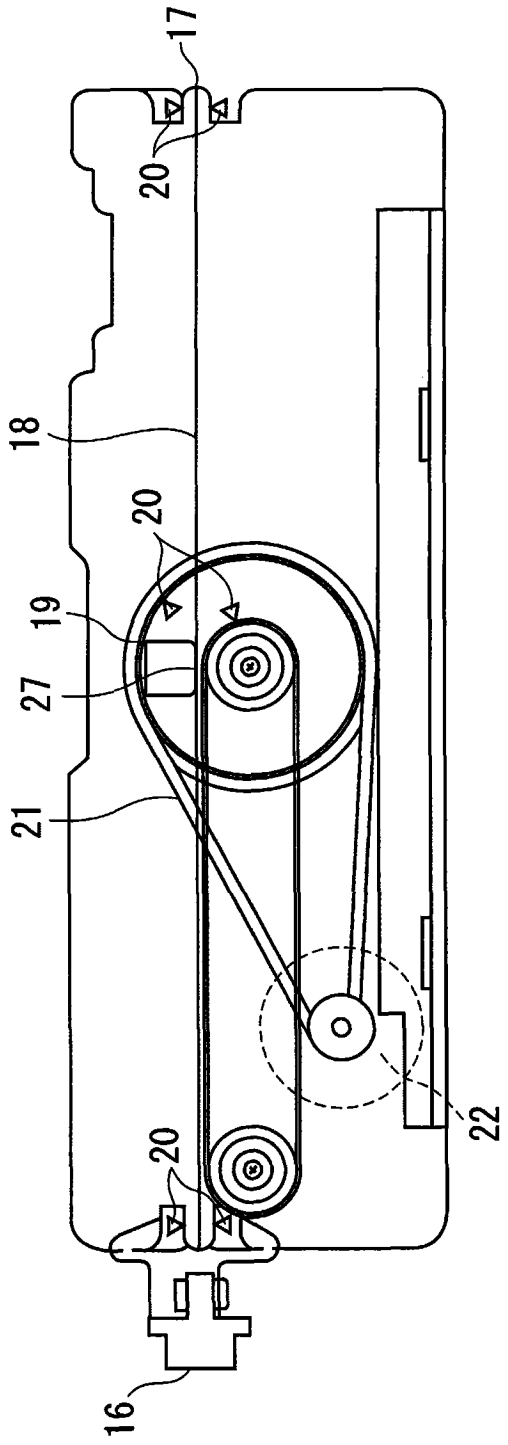



图 2

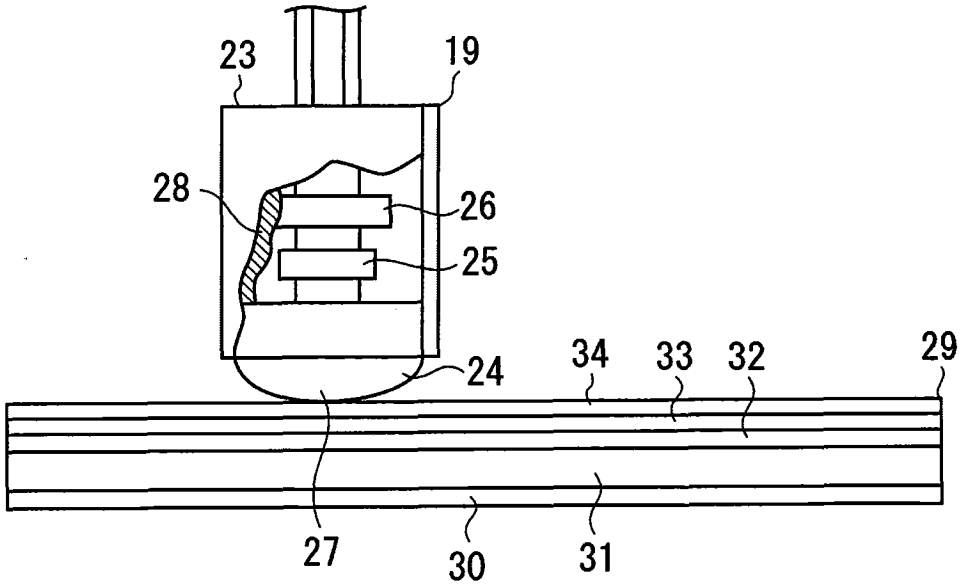

23
19
28
26
25
24
34
33
32
29
27
30
31

图 3

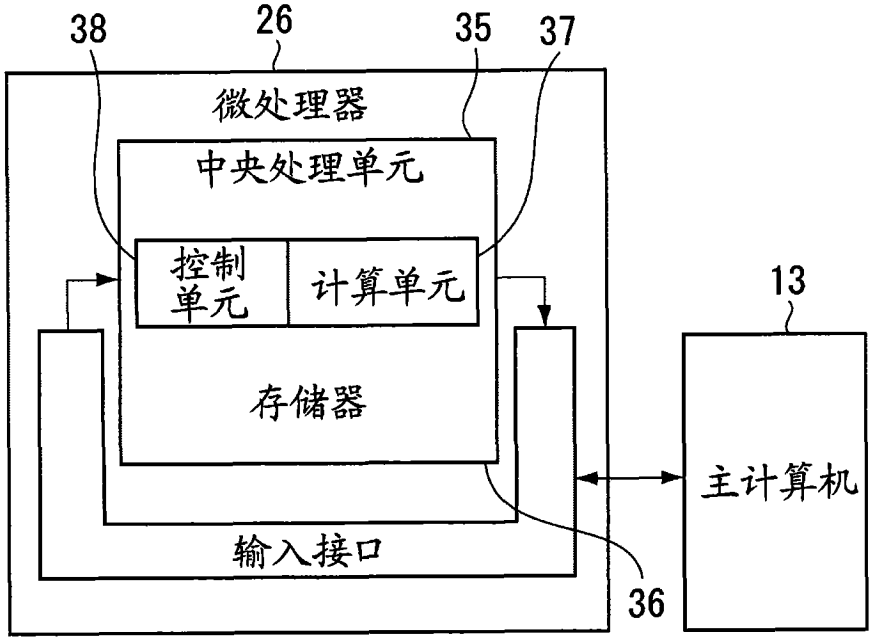

38
26
35
37
微处理器
中央处理单元
控制单元
计算单元
存储器
13
主计算机
输入接口
36

图 4

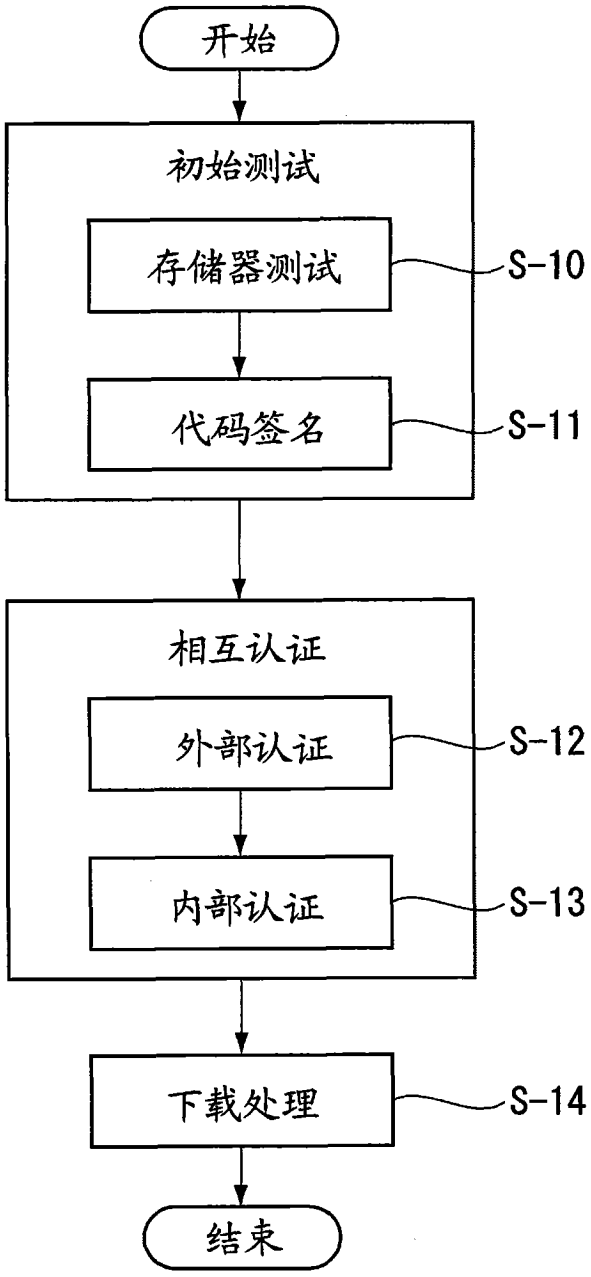



图 5

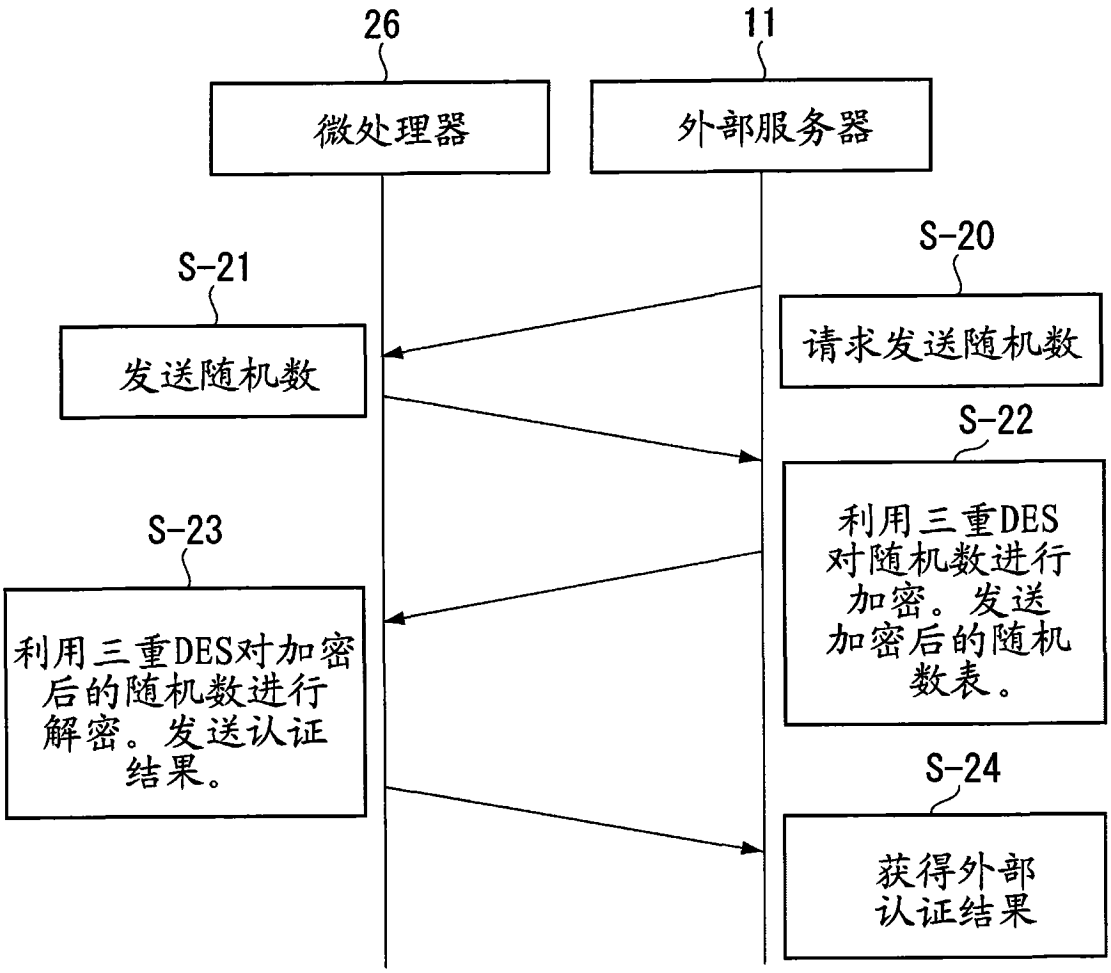

26
11
微处理器
外部服务器
S-20
S-21
请求发送随机数
发送随机数
S-22
利用三重DES对随机数进行加密。发送加密后的随机数表。
S-23
利用三重DES对加密后的随机数进行解密。发送认证结果。
S-24
获得外部认证结果

图 6

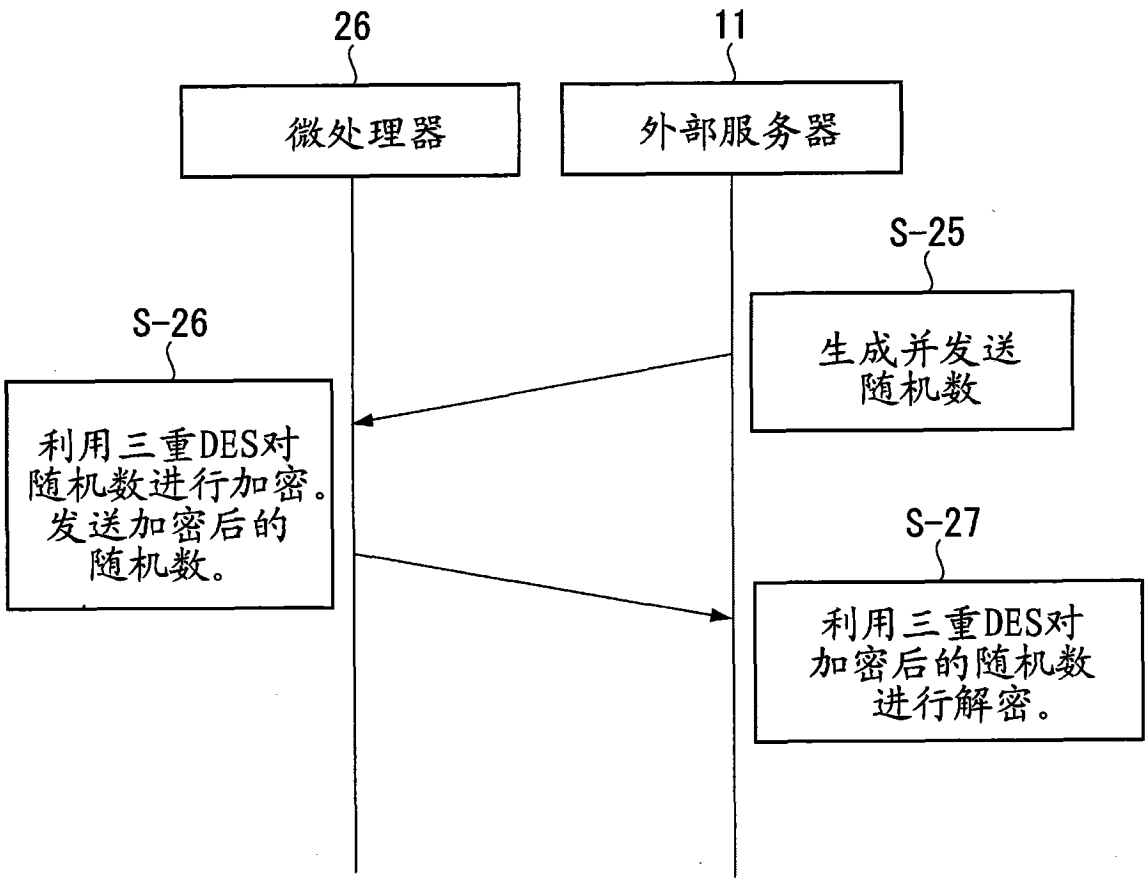

26
11
微处理器
外部服务器
S-25
生成并发送
随机数
S-26
利用三重DES对
随机数进行加密。
发送加密后的
随机数。
S-27
利用三重DES对
加密后的随机数
进行解密。

图 7

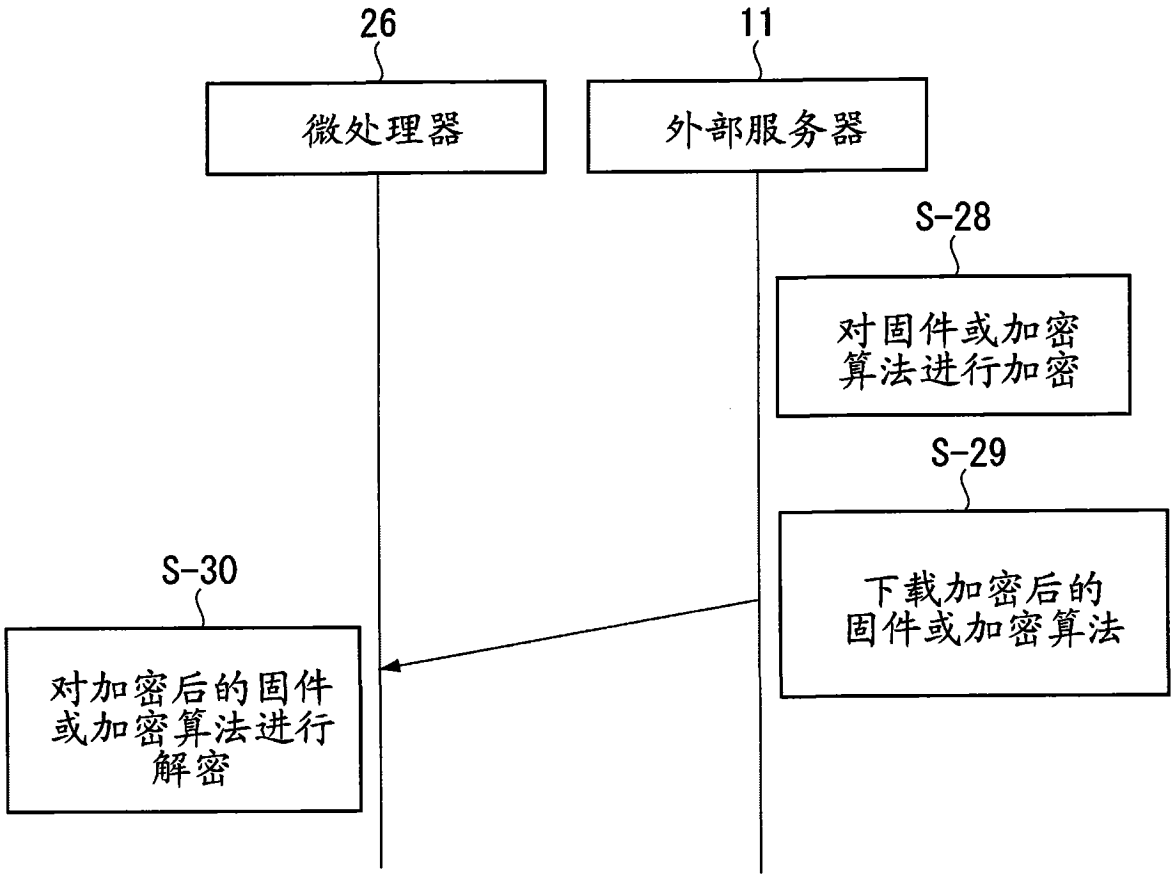

26
11
微处理器
外部服务器
S-28
对固件或加密算法进行加密
S-29
下载加密后的固件或加密算法
S-30
对加密后的固件或加密算法进行解密

图 8

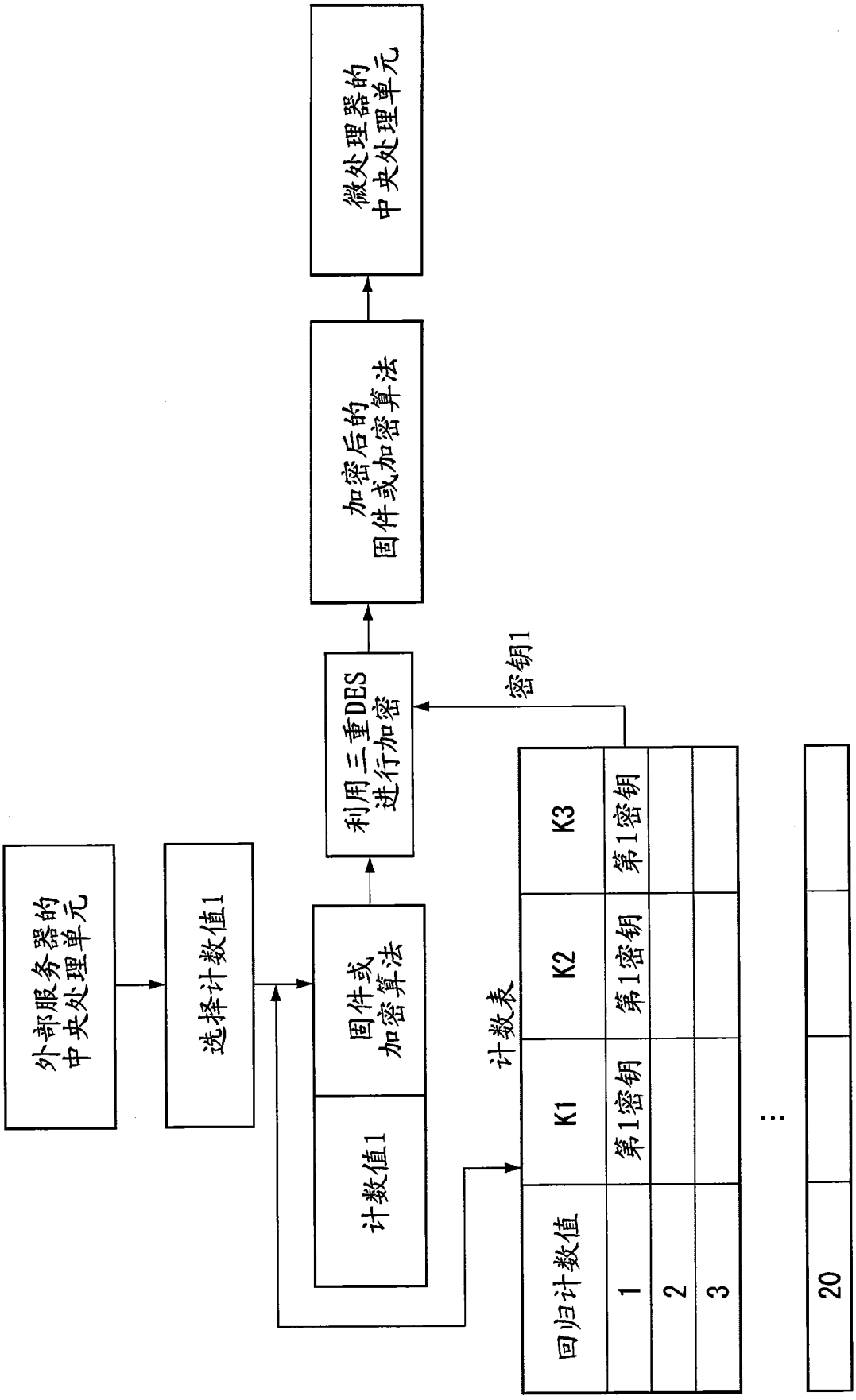

外部服务器的
中央处理单元
选择计数值1
固件或
加密算法
计数值1
利用三重DES
进行加密
密钥1
加密后的
固件或加密算法
微处理器的
中央处理单元
计数表
回归计数值
K1
K2
K3
1
第1密钥
第1密钥
第1密钥
2
3
20

图 9

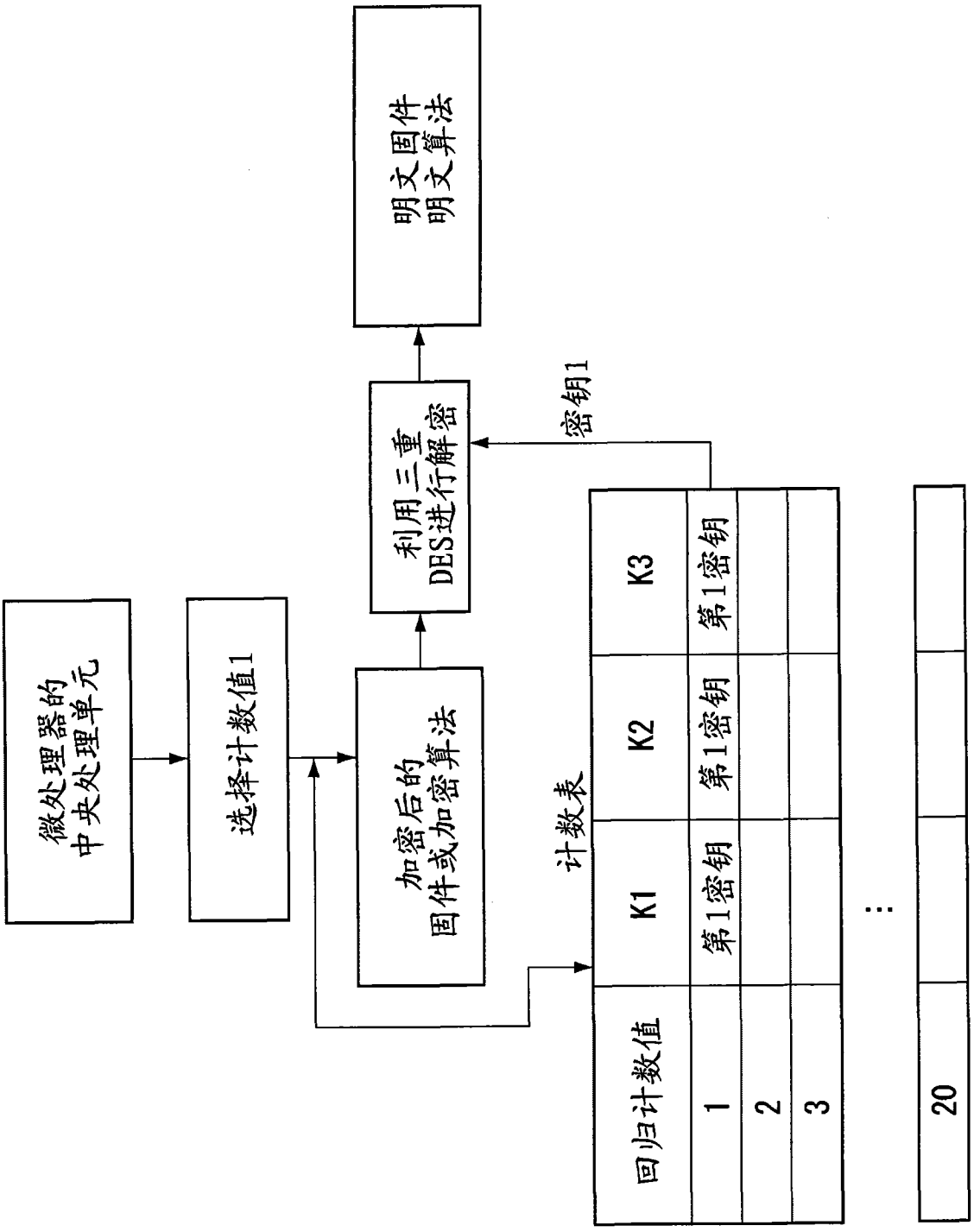

微处理器的
中央处理单元
选择计数值1
加密后的
固件或加密算法
利用三重
DES进行解密
明文固件
明文算法
密钥1
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第1密钥
第1密钥

图 10

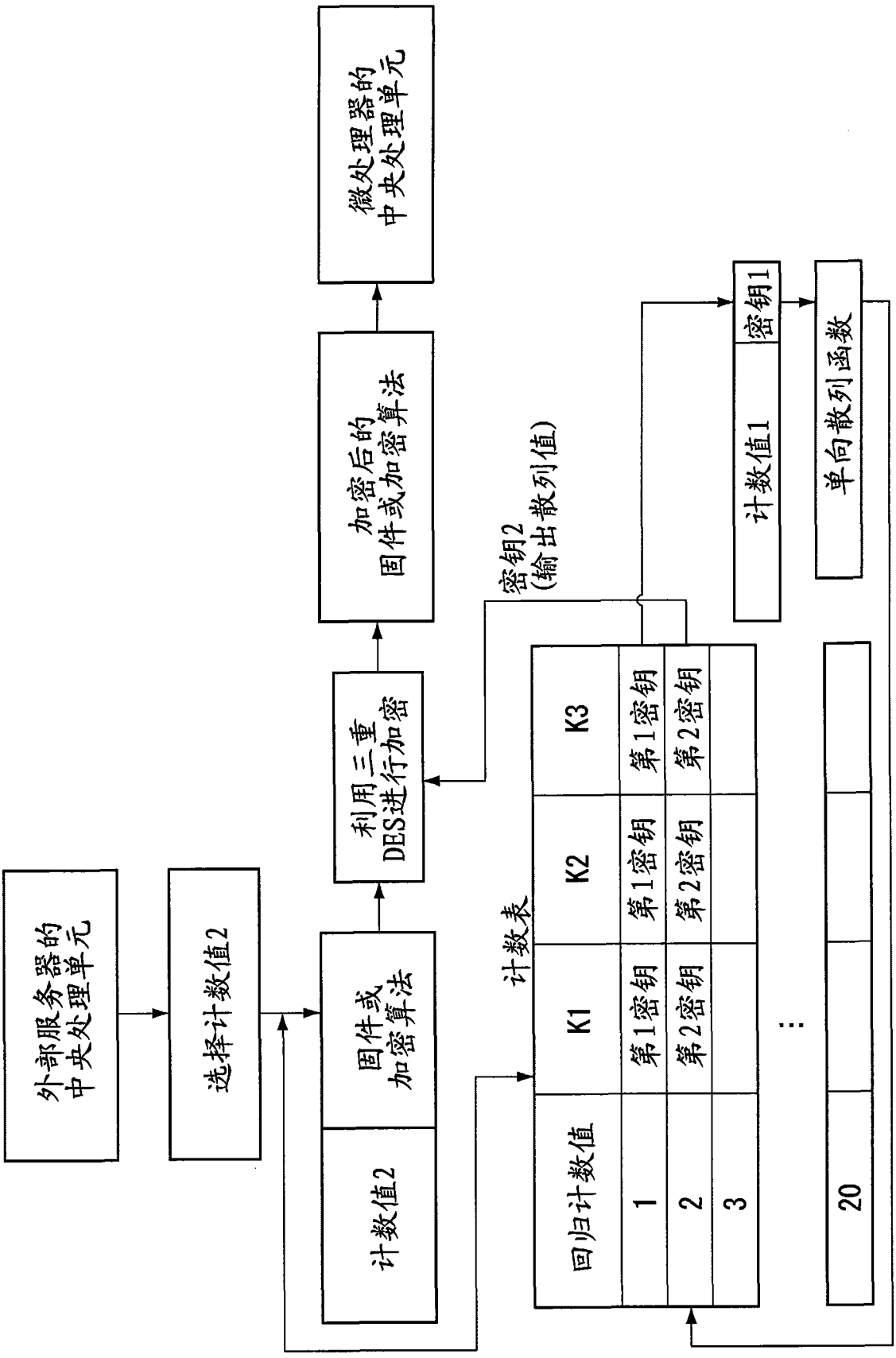

外部服务器的中央处理单元
选择计数值2
计数值2
固件或加密算法
利用三重DES进行加密
加密后的固件或加密算法
微处理器的中央处理单元
密钥2(输出散列值)
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第2密钥
第1密钥
第2密钥
第1密钥
第2密钥
计数值1
密钥1
单向散列函数

图 11

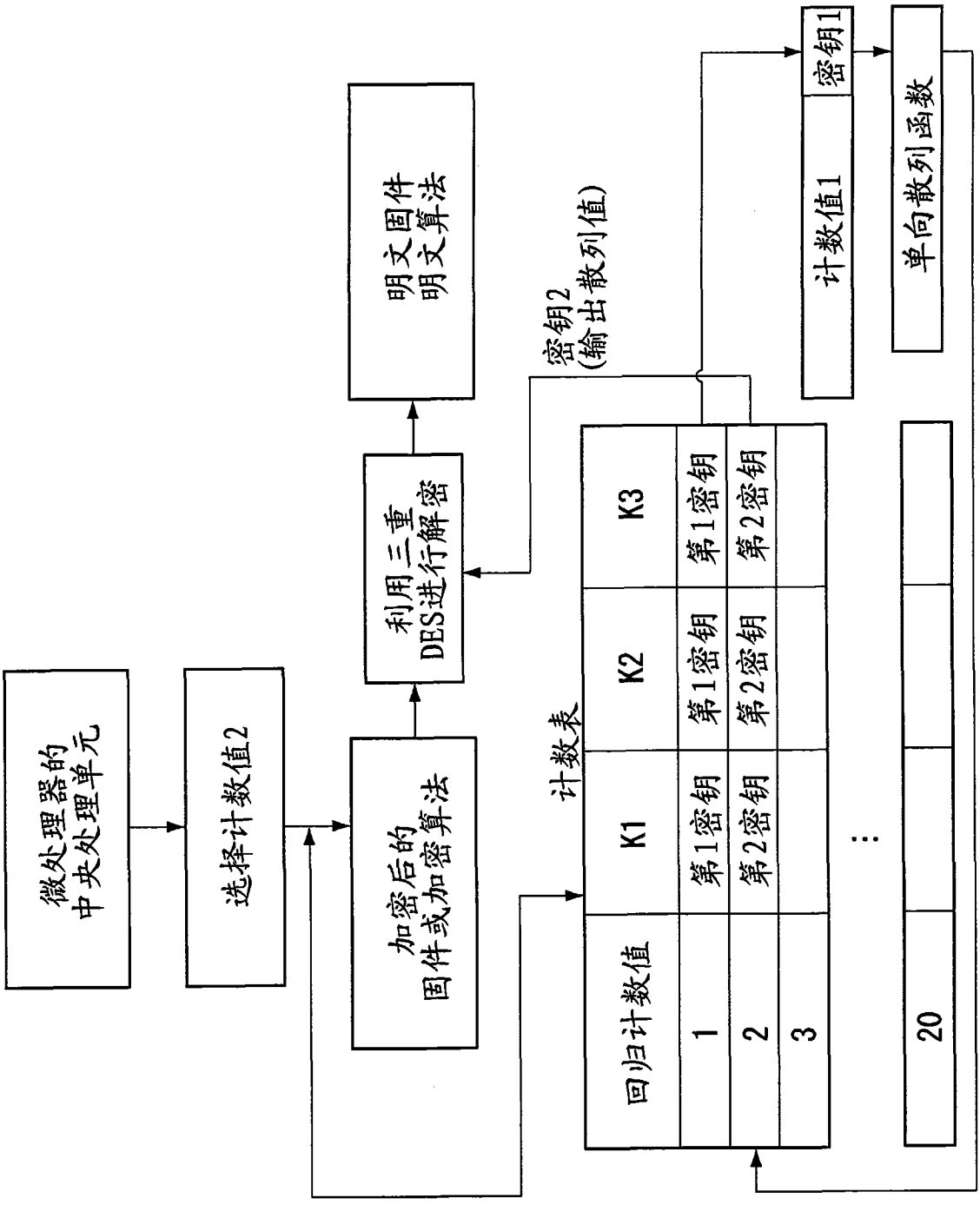

微处理器的
中央处理单元
选择计数值2
加密后的
固件或加密算法
利用三重
DES进行解密
明文固件
明文算法
密钥2
(输出散列值)
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第2密钥
计数值1
密钥1
单向散列函数

图 12

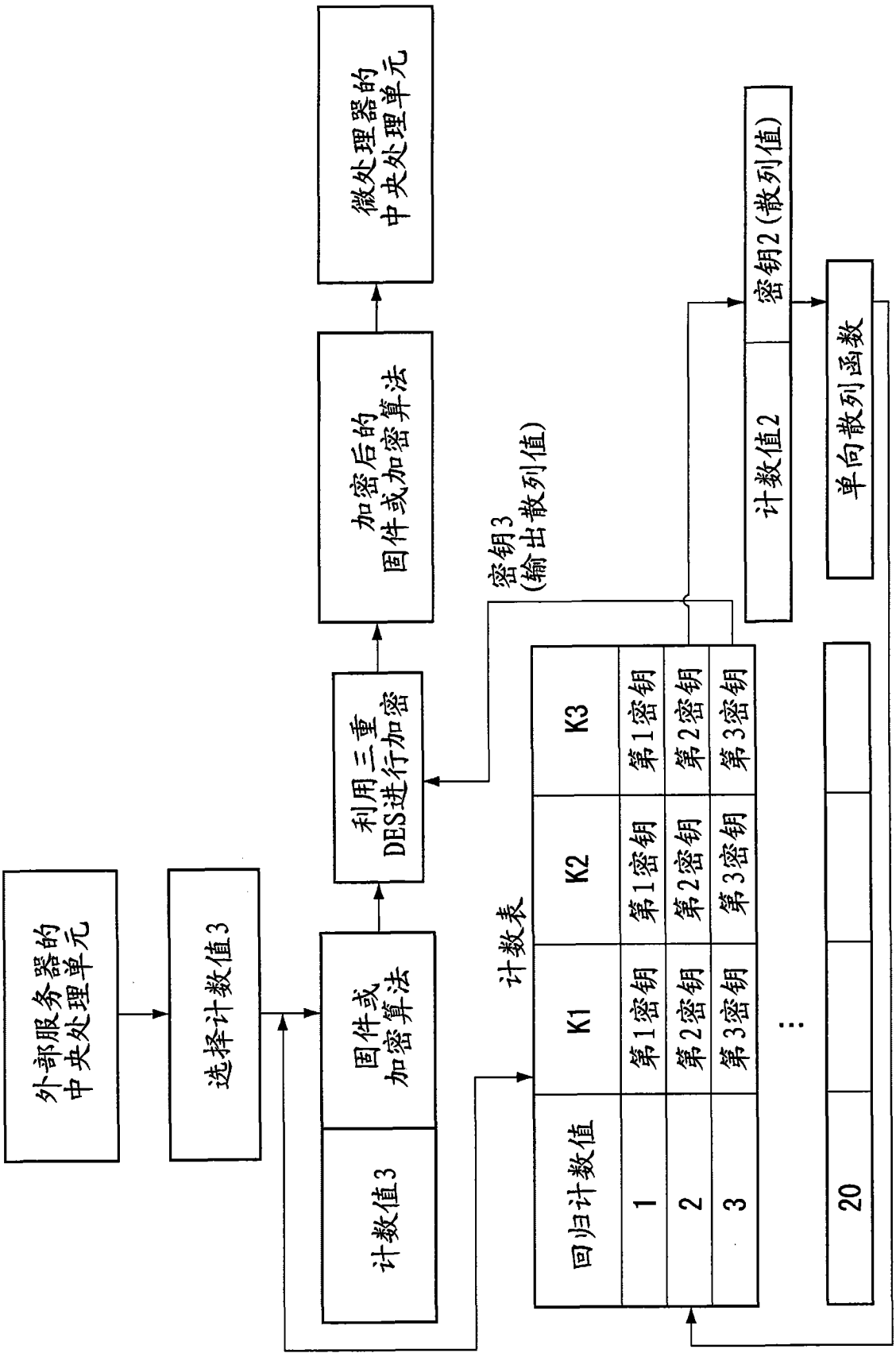

外部服务器的
中央处理单元
选择计数值3
固件或
加密算法
计数值3
利用三重
DES进行加密
加密后的
固件或加密算法
微处理器的
中央处理单元
密钥3
(输出散列值)
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第2密钥
第3密钥
计数值2
密钥2(散列值)
单向散列函数

图 13

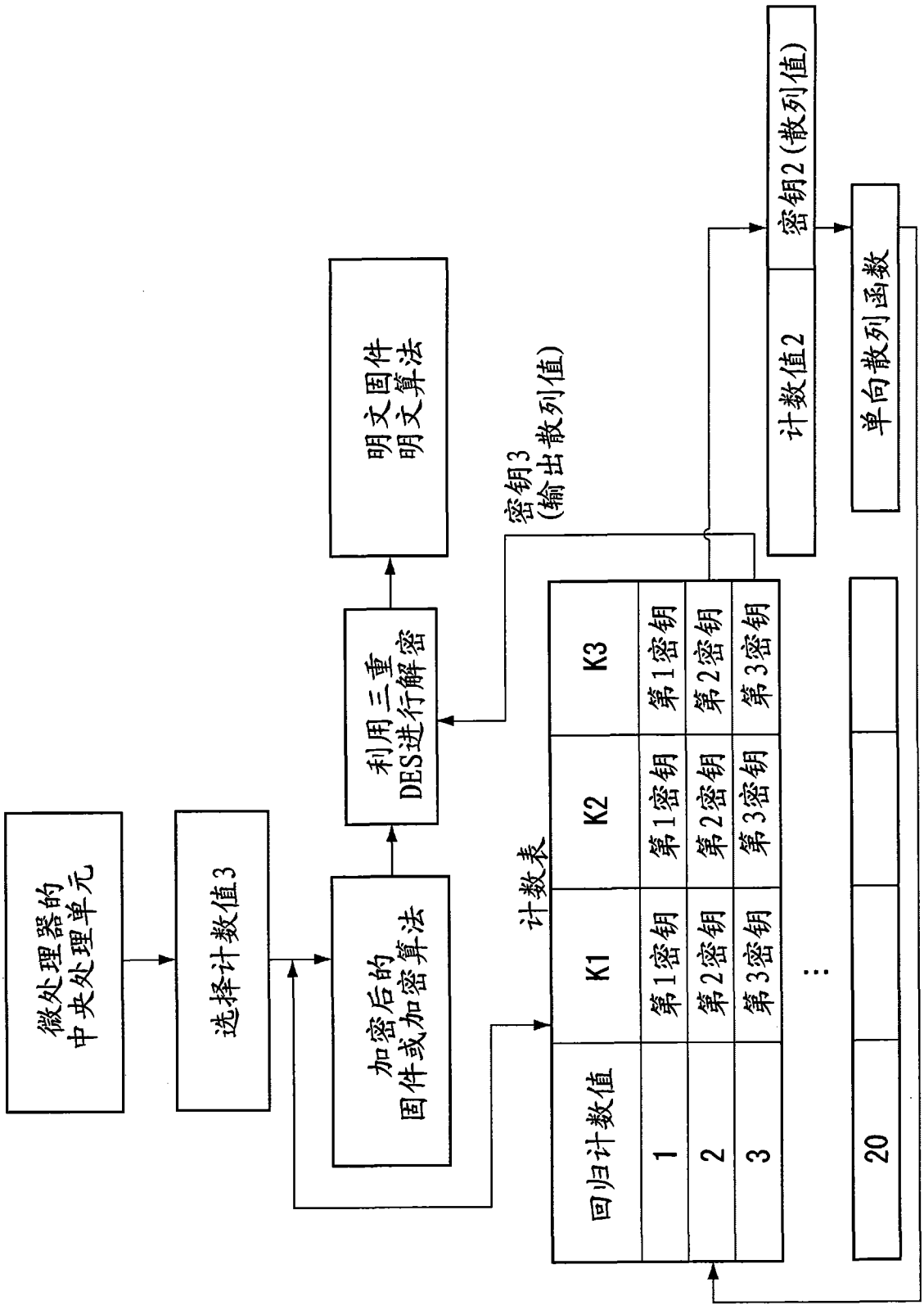

微处理器的
中央处理单元
选择计数值3
加密后的
固件或加密算法
利用三重
DES进行解密
明文固件
明文算法
密钥3
(输出散列值)
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第2密钥
第3密钥
计数值2
密钥2(散列值)
单向散列函数

图 14

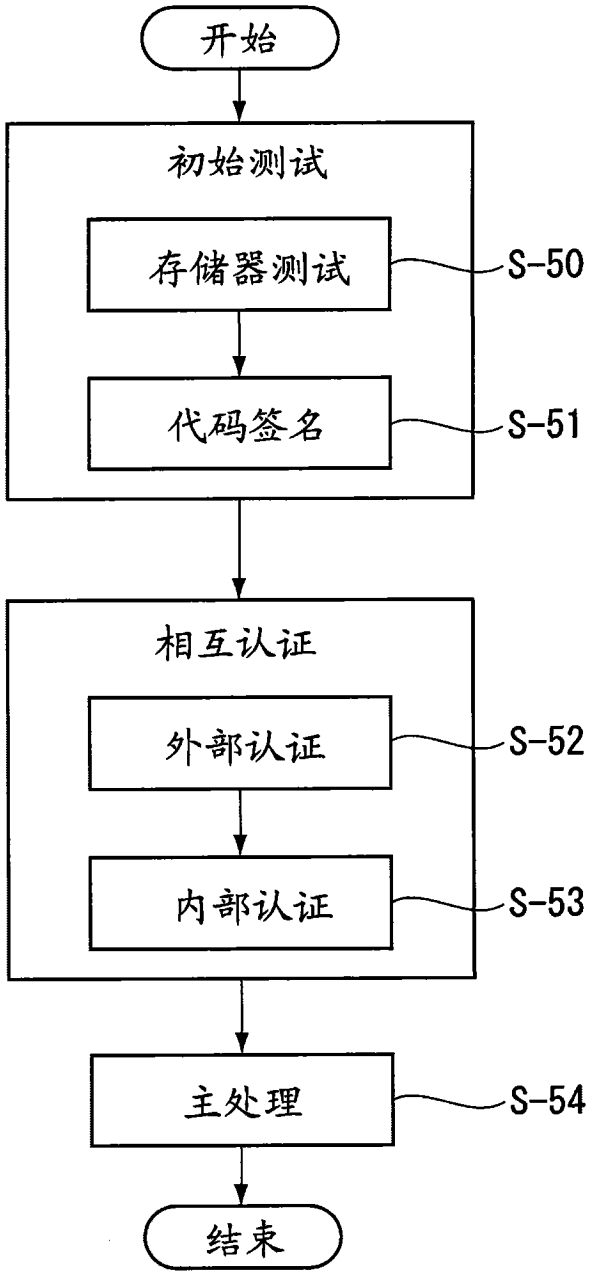



图 15

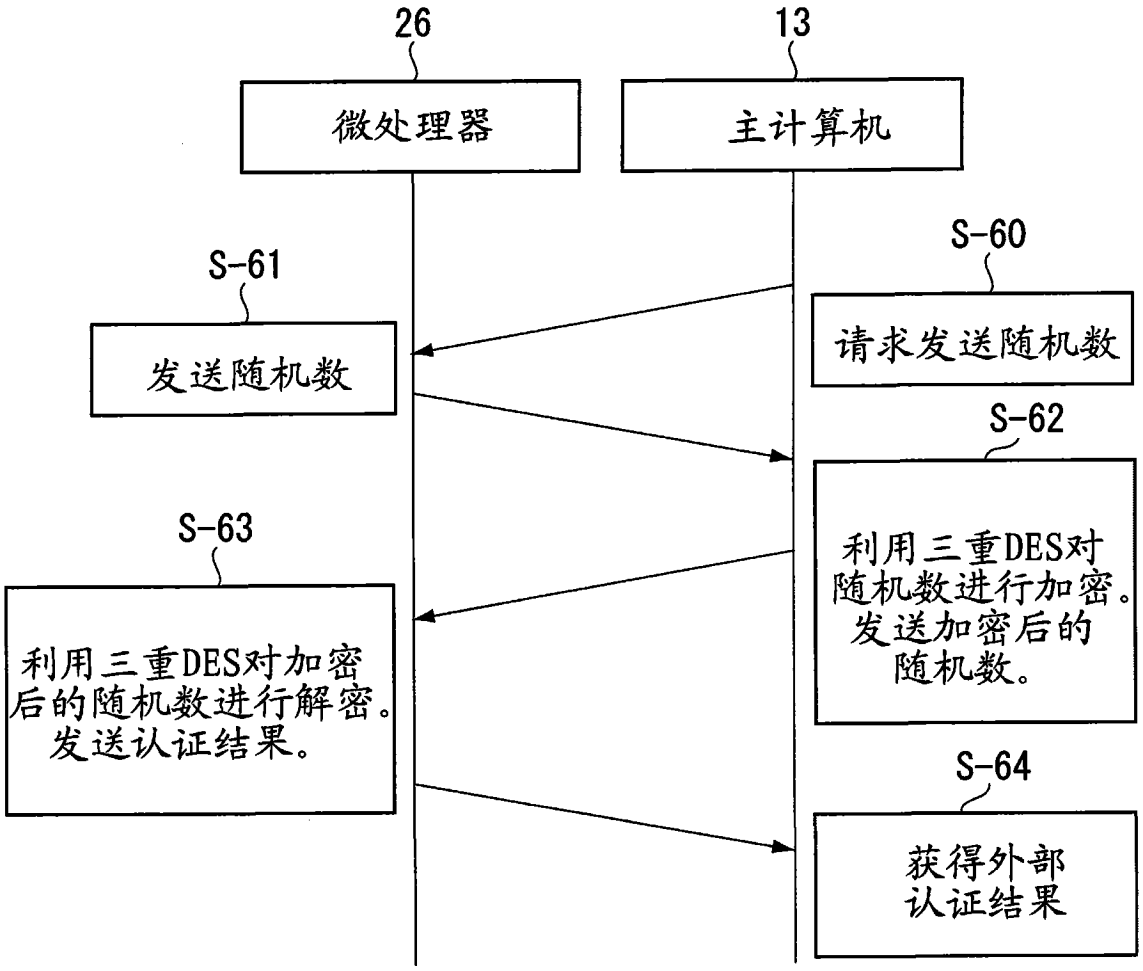

26
13
微处理器
主计算机
S-60
S-61
请求发送随机数
发送随机数
S-62
S-63
利用三重DES对随机数进行加密。发送加密后的随机数。
利用三重DES对加密后的随机数进行解密。发送认证结果。
S-64
获得外部认证结果

图 16

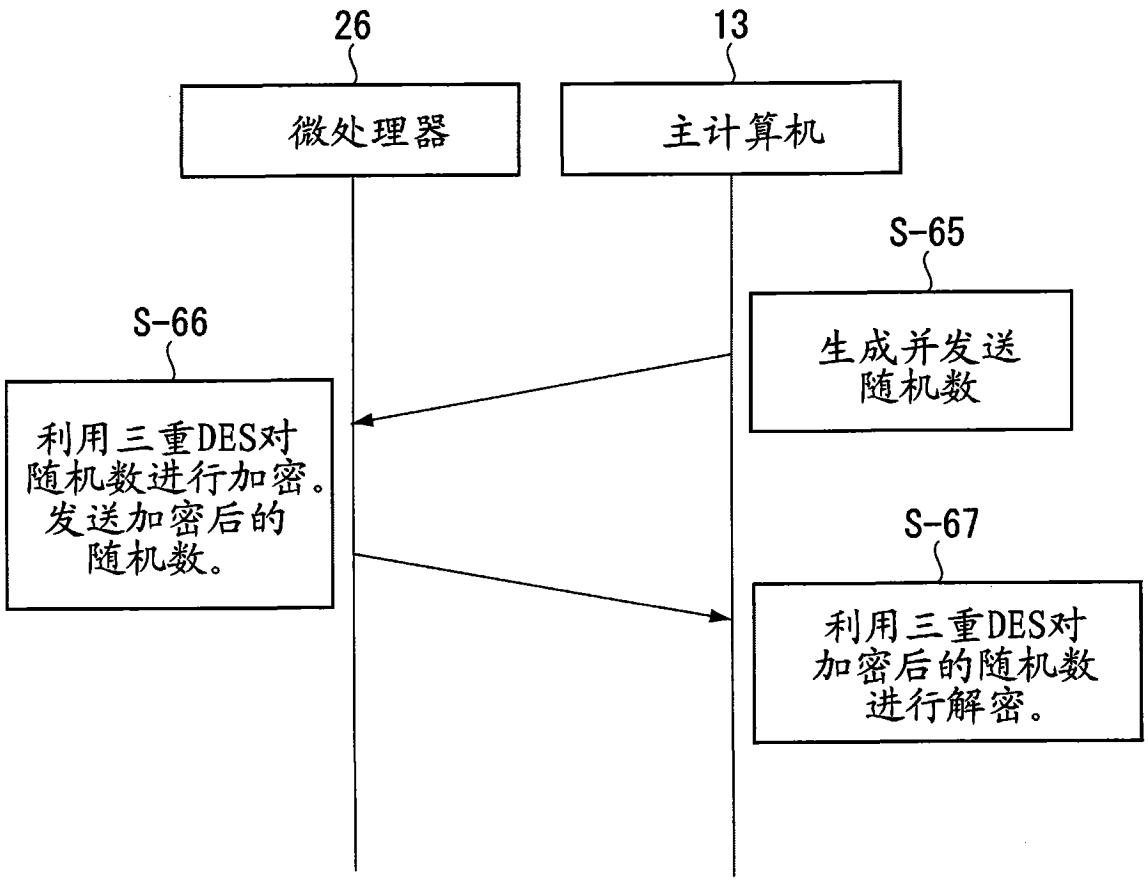

26
13
微处理器
主计算机
S-65
生成并发送
随机数
S-66
利用三重DES对
随机数进行加密。
发送加密后的
随机数。
S-67
利用三重DES对
加密后的随机数
进行解密。

图 17

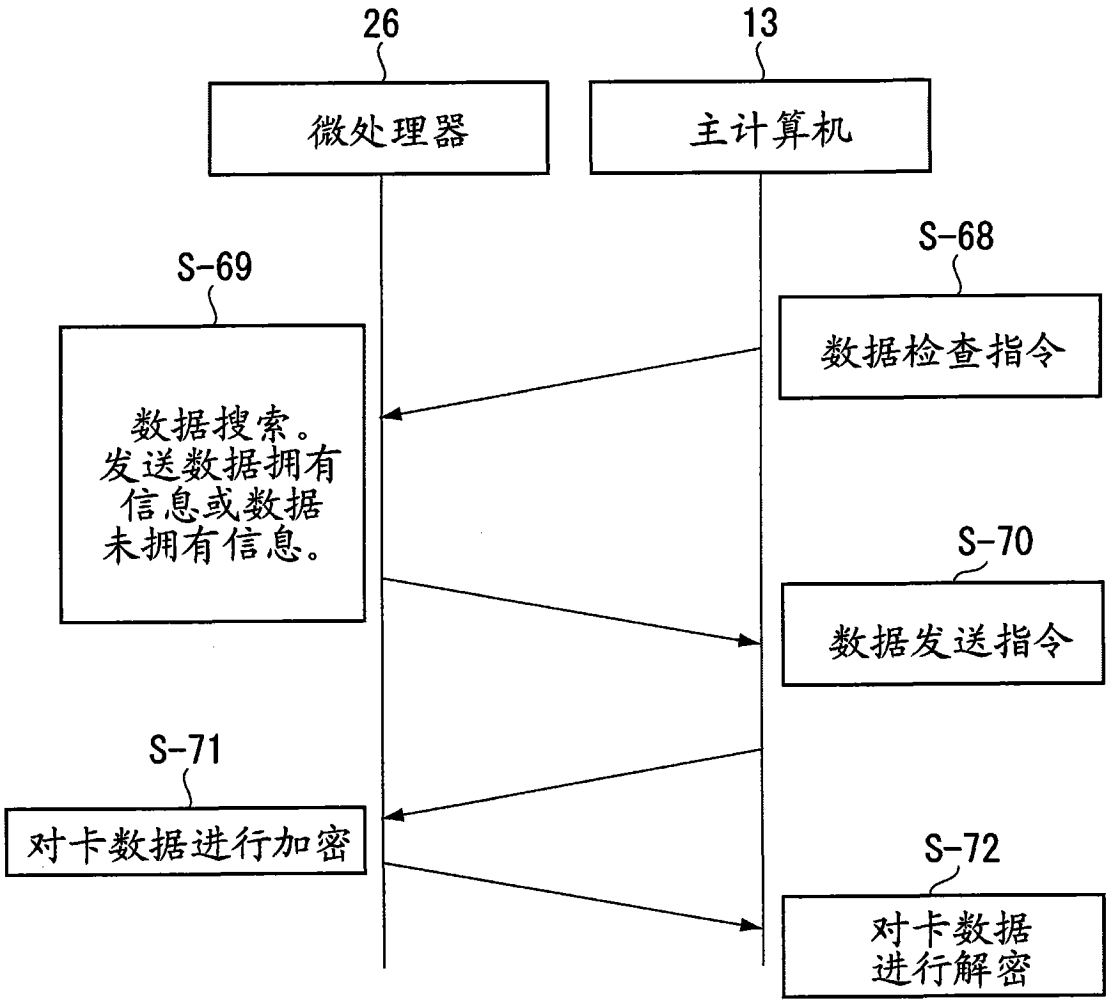

26
13
微处理器
主计算机
S-68
S-69
数据检查指令
数据搜索。
发送数据拥有
信息或数据
未拥有信息。
S-70
数据发送指令
S-71
对卡数据进行加密
S-72
对卡数据
进行解密

图 18

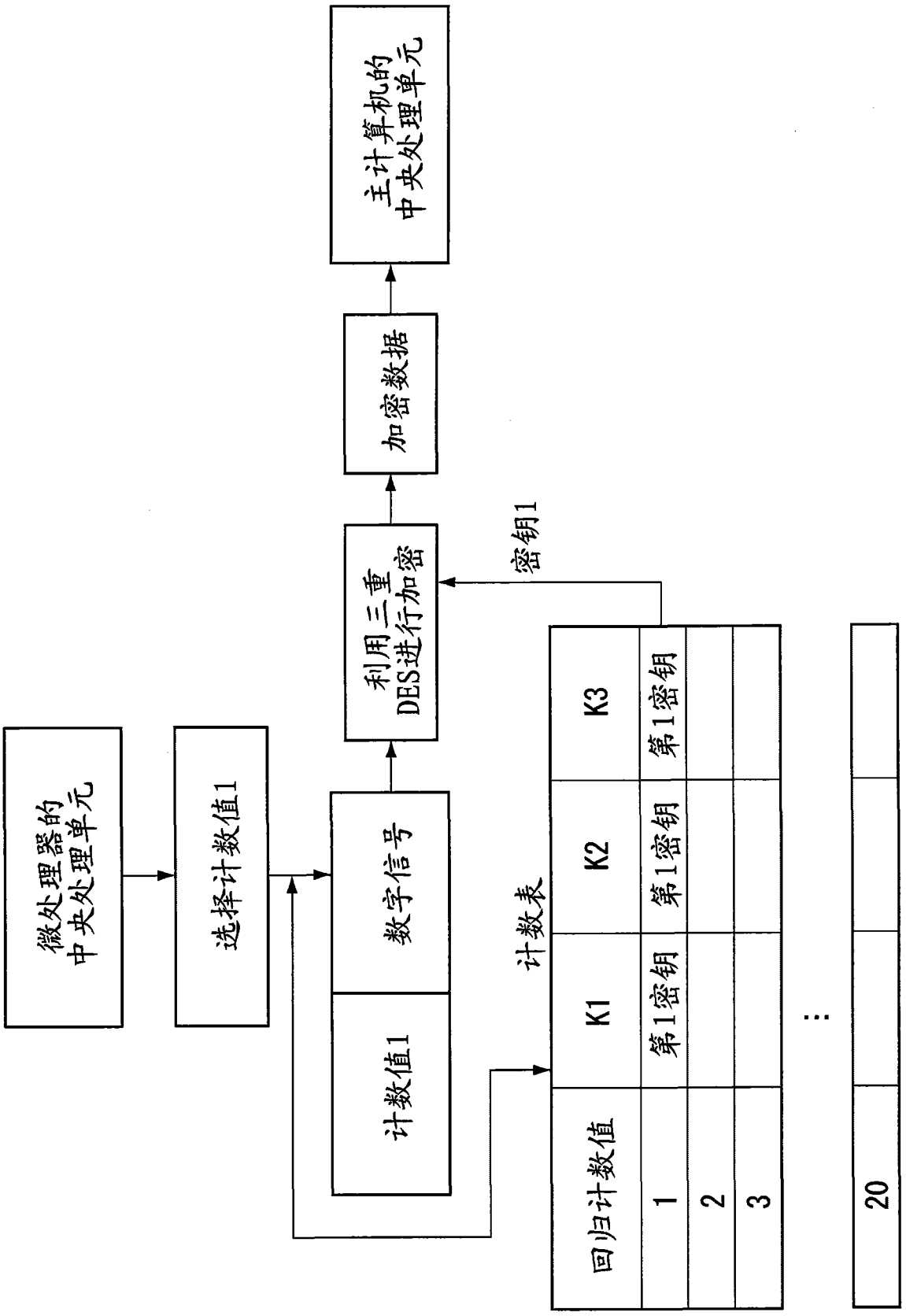

主计算机的中央处理单元
加密数据
利用三重DES进行加密
密钥1
微处理器的中央处理单元
选择计数值1
数字信号
计数值1
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第1密钥
第1密钥

图 19

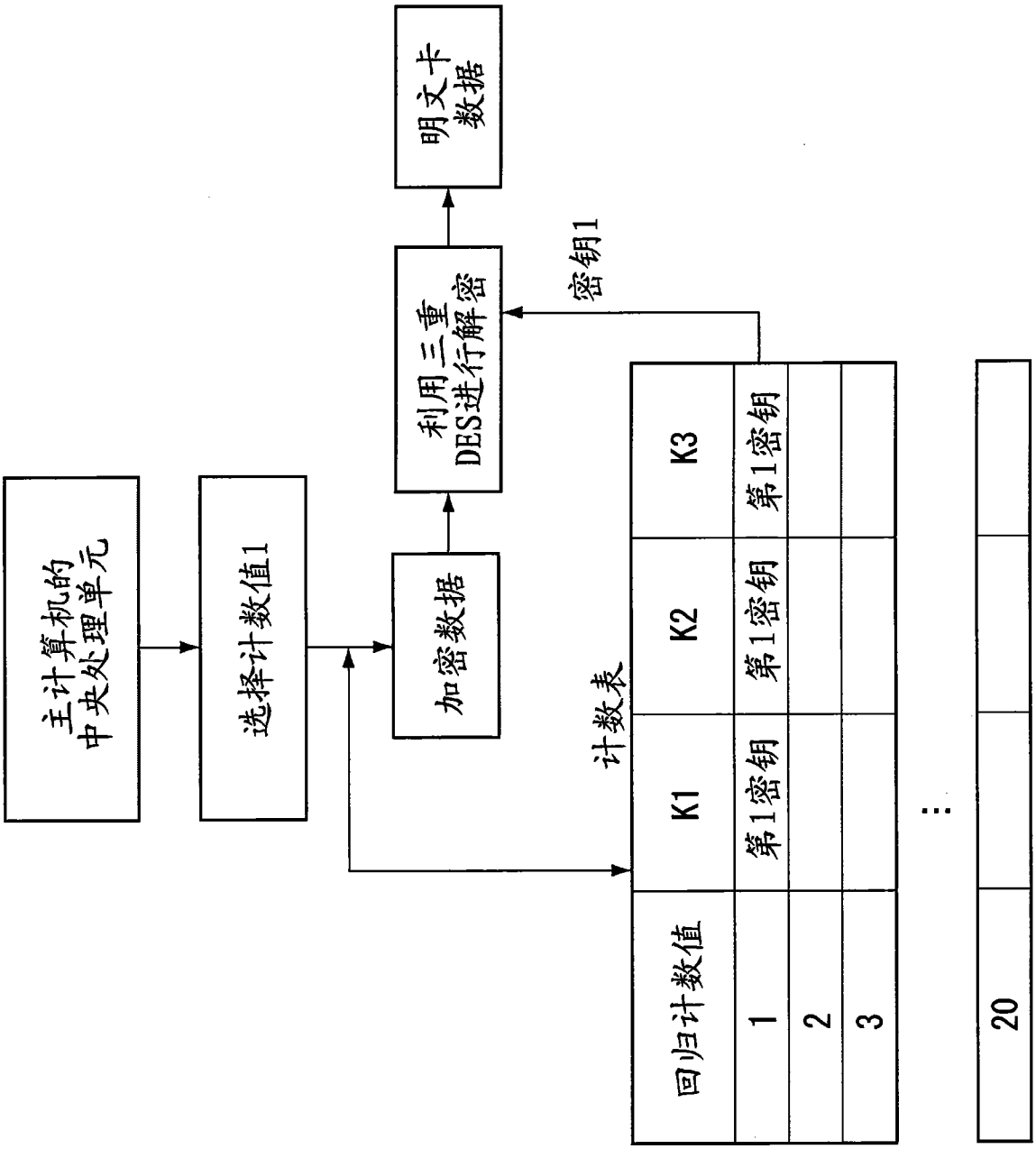

主计算机的
中央处理单元
选择计数值1
加密数据
利用三重
DES进行解密
明文卡
数据
密钥1
计数表
回归计数值
K1
K2
K3
1
第1密钥
第1密钥
第1密钥
2
3
20

图 20

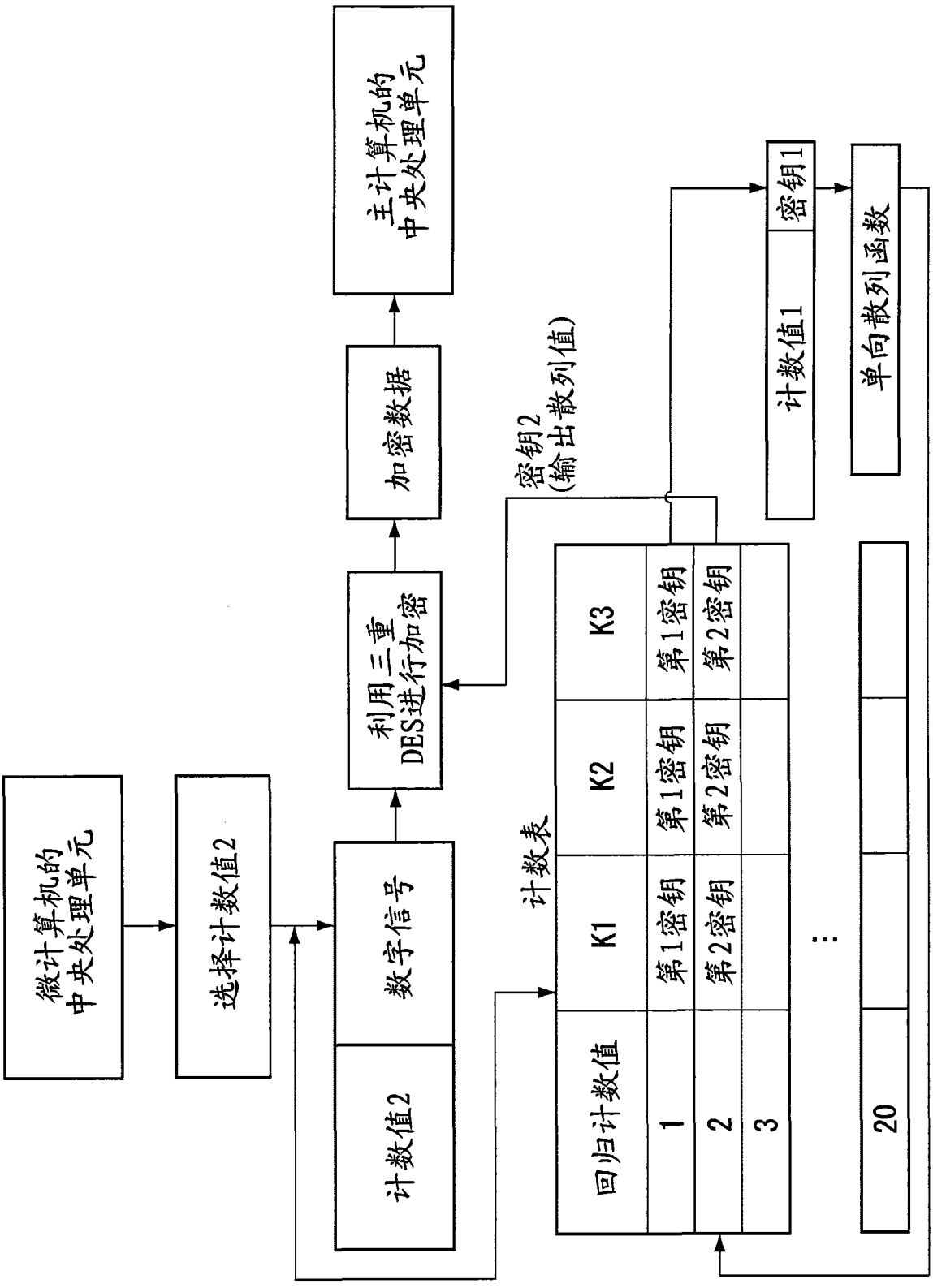

微计算机的中央处理单元
选择计数值2
数字信号
计数值2
利用三重DES进行加密
加密数据
主计算机的中央处理单元
密钥2(输出散列值)
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第2密钥
计数值1
密钥1
单向散列函数

图 21

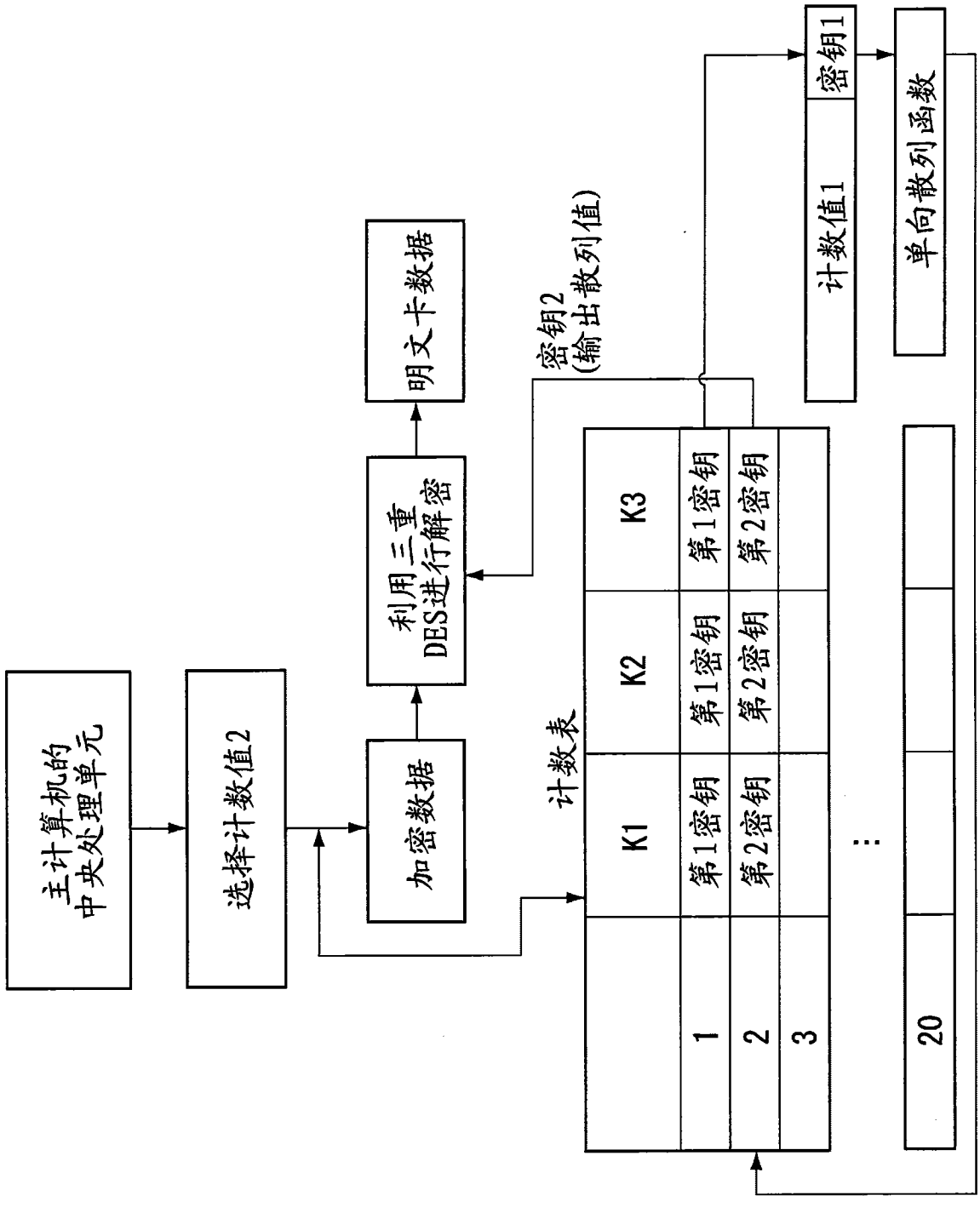

主计算机的
中央处理单元
选择计数值2
加密数据
利用三重
DES进行解密
明文卡数据
密钥2
(输出散列值)
计数表
K1
K2
K3
第1密钥
第2密钥
1
2
3
20
计数值1
密钥1
单向散列函数

图 22

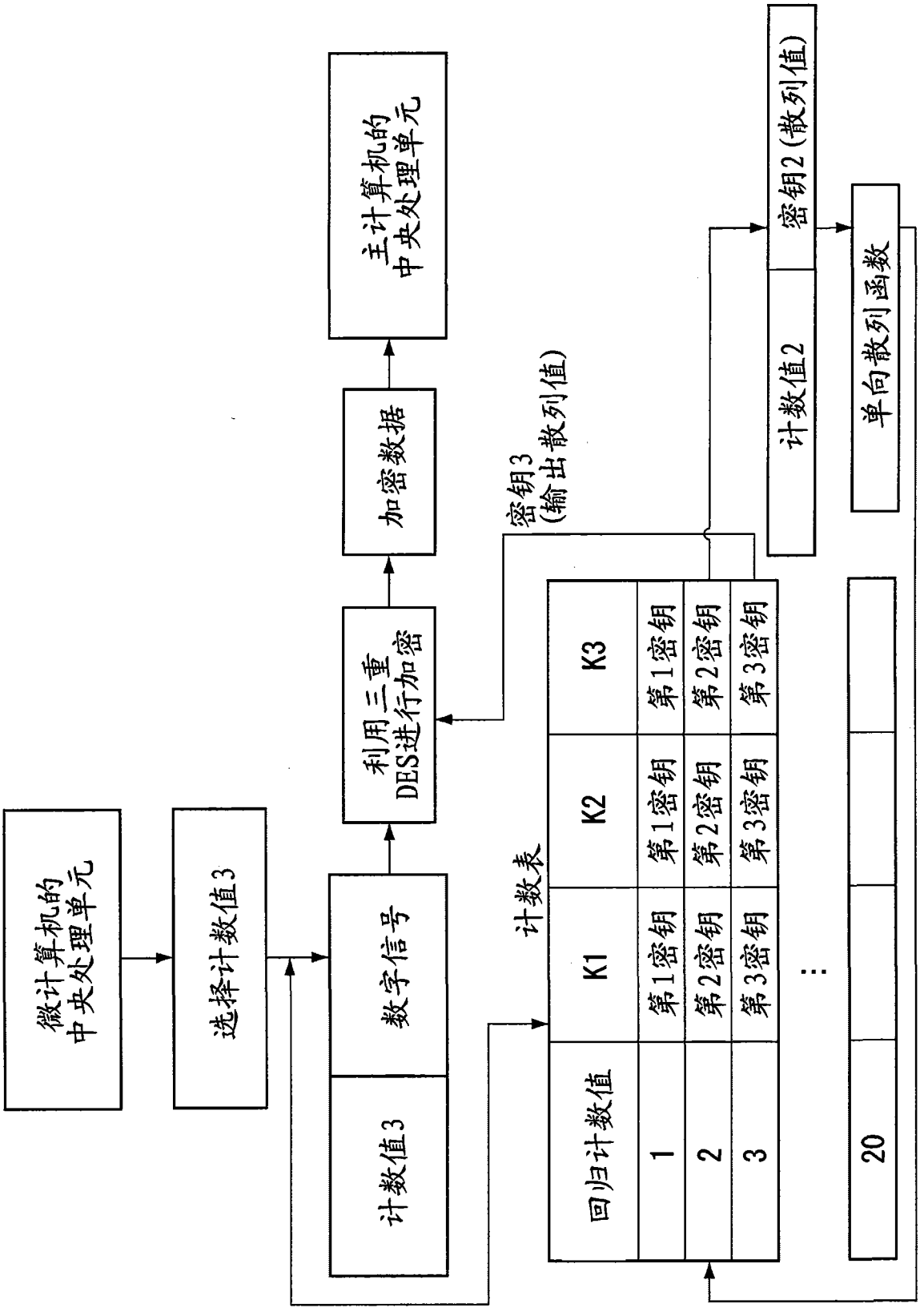

微计算机的
中央处理单元
选择计数值3
数字信号
计数值3
利用三重
DES进行加密
加密数据
主计算机的
中央处理单元
密钥3
(输出散列值)
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第2密钥
第3密钥
计数值2
密钥2 (散列值)
单向散列函数

图 23

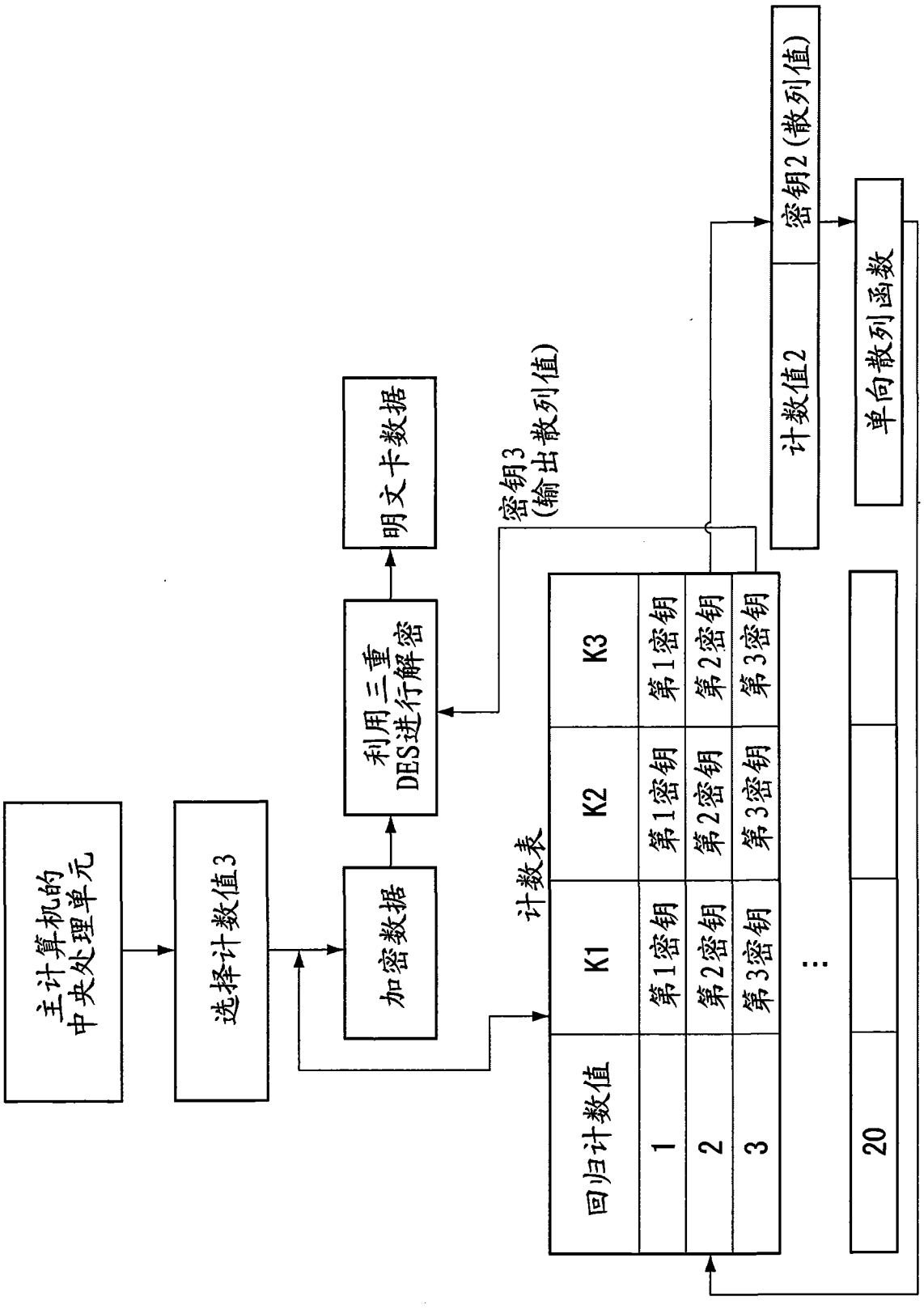

主计算机的
中央处理单元
选择计数值3
加密数据
利用三重
DES进行解密
明文卡数据
密钥3
(输出散列值)
计数表
回归计数值
K1
K2
K3
1
2
3
20
第1密钥
第2密钥
第3密钥
计数值2
密钥2(散列值)
单向散列函数

图 24